NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

CONDUCT AND ASSESSMENT OF A2C2 EXPERIMENT 3 AND GUIDELINES FOR FUTURE EXPERIMENTATION

by

Robert E. L. Benson

June, 1998

Thesis Co-Advisors: William G. Kemple
Gary R. Porter

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1998	Master's Thesis

4. TITLE AND SUBTITLE
CONDUCT AND ASSESSMENT OF A2C2 EXPERIMENT 3 AND GUIDELINES FOR FUTURE EXPERIMENTATION

5. FUNDING NUMBERS

6. AUTHOR(S)
Benson, Robert E. L.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The Adaptive Architectures for Command and Control (A2C2) project is sponsored by the Office of Naval Research (ONR) and is focused on analysis of joint decision-making at the operational level and adaptation of joint command and control architectures. To accomplish this objective, the A2C2 project team has conducted a series of human-in-the-loop experiments at the Naval Postgraduate School (NPS). The third experiment of the series was conducted during November 1997. This experiment differed from previous A2C2 experiments in that it focused on how organizations adapt their structures to maximize their effectiveness under changing events. This thesis reports on the planning and conduct of Experiment 3 with a focus on the contributions made by author and the Lead Team of officer-students and the analysis of their hypotheses. The author examines data collected during Experiment 3 in support of these hypotheses. A detailed statistical analysis is performed and results discussed. Finally, a discussion of lessons learned from the author's perspective pertaining to the experiment is given along with recommendations for conducting future experiments at NPS.

14. SUBJECT TERMS
Adaptive Architectures for Command and Control (A2C2),
Statistical Analysis of Measures of Effectiveness,
Human-in-the-Loop Experimentation at NPS

15. NUMBER OF PAGES
127

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFI- CATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

CONDUCT AND ASSESSMENT OF A2C2 EXPERIMENT 3 AND GUIDELINES FOR FUTURE EXPERIMENTATION

Robert E. L. Benson
Captain, United States Marine Corps
B.S., United States Naval Academy, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY (COMMAND, CONTROL, AND COMMUNICATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
June 1998

Author: ______________________
Robert E. L. Benson

Approved by: ______________________
William G. Kemple, Thesis Advisor

____
Gary R. Porter, Thesis Co-Advisor

____
Dan C. Boger, Chairman,
C4I Academic Group

ABSTRACT

The Adaptive Architectures for Command and Control (A2C2) project is sponsored by the Office of Naval Research (ONR) and is focused on analysis of joint decision-making at the operational level and adaptation of joint command and control architectures. To accomplish this objective, the A2C2 project team has conducted a series of human-in-the-loop experiments at the Naval Postgraduate School (NPS). The third experiment of the series was conducted during November 1997. This experiment differed from previous A2C2 experiments in that it focused on how organizations adapt their structures to maximize their effectiveness under changing events. This thesis reports on the planning and conduct of Experiment 3 with a focus on the contributions made by author and the Lead Team of officer-students and the analysis of their hypotheses. The author examines data collected during Experiment 3 in support of these hypotheses. A detailed statistical analysis is performed and results discussed. Finally, a discussion of lessons learned from the author's perspective pertaining to the experiment is given along with recommendations for conducting future experiments at NPS.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The Adaptive Architectures for Command and Control (A2C2) research project is an ambitious four-year endeavor in model-based experimentation sponsored by the Office of Naval Research (ONR). The program is a distributed effort among industry, university, and government researchers whose goals include: 1) extending 12 years of naval composite warfare decision-making research into the joint Command and Control (C2) arena; 2) focusing on adaptive architectures within decision-making organizations; and 3) producing results that range from the purely theoretical to those that can be used by operational forces. The A2C2 experiment design combines an operational scenario with computer-based architecture models and tests these architectures in a series of human-in-the-loop experiments using military officers operating in a Joint setting as the test subjects. Results from each experiment, as well as lessons learned, are then applied toward the next experiment. As of the writing of this thesis, three experiments have been successfully completed at NPS.

The first experiment in A2C2 was concluded in March 1996 and was designed to be a test run of the simulation software and lay the foundation for future experiments. Experiment 2 was conducted in November 1996 and built on the results of the first experiment by exploring the interaction of task structure with organizational structure and the drivers of adaptation (workload, uncertainty, coordination demands, and structure topology). Experiment 3 advances the project objectives by testing specific research hypotheses concerning

incremental adaptation of organizational structures. Specifically, it was desired to explore the theory that organizations tend to adapt, when forced by external events, in small steps rather than a large leap toward a supposedly optimal architecture. Other objectives of Experiment 3 focused on comparing experimental data with pre-experiment model predictions and learning more about the Joint C2 decision-making processes.

Contributing to the NPS effort in the A2C2 project was a Lead Team of officer-students who were tasked both with performing a variety of support and administrative roles before and during the experiment and crafting their own research questions for which data could be collected and analyzed and that could be answered within the experiment design. The author of this thesis was the Lead Team leader and coordinated their efforts. The focus of this thesis is the analysis and conclusions of the Lead Team's hypotheses concerning the relationship between task performance and model-based architectures. Additionally, the possibility that excessive learning occurred throughout the experiment runs was explored. The four Lead Team objectives concerned:

1. The relationship between the number of decision-making nodes per task and architecture.
2. The average accuracy per task versus architecture.
3. The average performance per task versus architecture.
4. The possibility that learning continued throughout the experiment trials.

Measures were developed and collected from the simulation software files compiled after each run. The analysis performed by the Lead Team was based on a single factor (architecture), which when varied would yield valuable data for

analysis. This factor was controlled at six levels corresponding to the six different architectures played.

For each of the teams, the experiment consisted of three scenario "runs". The teams conducted the first practice run and then were presented with a "trigger". The trigger was designed to force the teams to adapt and was presented in the form of a severe reduction (~30%) in the number of assets available to the teams for the following two runs without change in mission tasks. The pre-trigger runs were conducted with a six-node architecture. The post-trigger runs were conducted with four-, five-, and six-node architectures. Though the trigger reduced the total assets available to the teams, the architecture modelers at UConn redistributed the remaining assets to the nodes in the post-trigger alternative architectures in a manner that reduced the requirement for inter-nodal coordination relative to the pre-trigger architecture (also with reduced assets). Implicit in this design is the assumption that reduced inter-nodal coordination would result in better performance. Overall, there were six different asset-architecture combinations and nine "player" teams conducting three runs each for a total of 27 runs (including pre-trigger).

The Lead Team analysis plan called for the use of parametric and non-parametric analysis of variance, regression and graphical analyses to examine the four objectives discussed above. Based on this analysis, it was concluded for the first hypothesis that the average number of nodes per task was statistically different across architectures and closely paralleled the modelers'

predictions. Likewise, the performance, as measured by the accuracy per task, differed between architectures. However, the performance of the "optimal" 4-node architecture which, due to the inherent lack of inter-nodal coordination requirements was predicted by the modelers to be the highest, actually yielded the lowest performance scores. Possible reasons for the difference in predicted and actual performance are inadequate training of the subjects and the perceived time pressure that the post-trigger architectures with reduced nodes required. In pursuing whether the teams continued to experience undue learning which affected performance in later runs, the Lead Team examined the accuracy of the tasks as the teams performed more runs. The results suggest that accuracy increased as more runs were accomplished. Finally, it was determined that the number of tasks completed increased as the teams did more runs which reflects a team becoming more familiar with the mission tasks and supports the theory that significant learning was occurring.

The thesis concludes with the author's lessons learned from Experiment 3. The lessons learned are presented as guidelines for future experiments conducted at NPS. Scheduling, simulation software modifications, player preparation and training, laboratory preparation, and methods of information distribution are addressed and suggestions made to facilitate a smoother execution of the A2C2 research experimentation process.

I. INTRODUCTION

The 21st century promises significant changes for the military and Department of Defense. Concepts like the Revolution in Military Affairs (RMA), Navy Smart Ship (SC-21), and precision strike are based on a smaller, more technologically advanced military that will be reliant on a more flexible, closely coupled command and control system to achieve Full Spectrum Dominance (Joint Warfighting Center, (JWC)), pg. 2, 1997).

In 1995, the Office of Naval Research (ONR) commissioned a four-year research project titled "Adaptive Architectures for Command and Control (A2C2)". Force reductions and technological innovation are affecting the strategic outlook of the nation's leadership. In the 1998 Defense Strategy, Secretary of Defense Cohen states that: "We will execute the strategy with superior military forces that fully exploit advances in technology by employing new operational concepts and organizational structures" (Cohen, 1998). The A2C2 project's goal is to advance the state of knowledge regarding decision-making in organizational settings. This includes understanding more about adapting an organization's structure for various tasks/missions and capabilities. In future conflicts, information superiority will facilitate task organization changes needed to respond to unexpected situations (JWC, p. 61, 1997).

In November 1997, under the A2C2 project, a research group at the Naval Postgraduate School (NPS) conducted the third A2C2 experiment using nine six-person player teams to examine JTF level decision-making processes and

performance and the team's propensity to adapt their organizational structure to changing operational stimuli (i.e., trigger events). The section on the Experimental Approach in this chapter describes the trigger and how it was used. In particular, research was focused on the post trigger event adaptation decisions in order to answer whether teams are more inclined to adapt to an architecture similar to the one they are familiar with (proximity) or one which is more radical in its structure but better suited to perform the tasks (optimality).

A. THE A2C2 EXPERIMENT TEAM

The A2C2 research team is an interdisciplinary effort pooling resources from a wide range of expertise. The team consisted of researchers from private industry (Aptima, Inc., Alphatech, Inc.), government institutions (ONR, NPS), and college campuses (Carnegie-Mellon (CMU), University of Connecticut (UConn), George Mason University (GMU), Michigan State University (MSU)), and other agencies.

Contributing to the NPS effort in the A2C2 project was a Lead Team of officer-students who were tasked both with performing a variety of support and administrative roles before and during the experiment and crafting their own research questions for which data could be collected and analyzed and that could be answered within the experiment design. The Lead Team developed the Operation Orders (Oporders), trained the subjects, set up and ran the simulation, monitored team planning sessions, and collected data both for the A2C2 researchers and the Lead Team's own analysis. The Lead Team also conducted

and reported on their own analysis, assisted in the A2C2 research team analysis and generated lessons-learned to help with the conduct of future experiments.

The Lead Team for the experiment consisted of the senior class from the Joint Command, Control, Coordination, Computers, and Intelligence (JC4I) Systems curriculum. These nine officers (0-3 to 0-4) provided a wide range of operational experience from which to draw on as subject matter experts. The Lead Team's analysis was conducted as part of NPS course CC 4103, C4I Systems Evaluation.

The author, as the leader of the Lead Team, contributed to the A2C2 research by coordinating the Lead Team's efforts before, during and after the experiment. The author also participated in the A2C2 project post-experiment data analysis. Finally, the author and the Lead Team developed lessons-learned from this experiment from a design and setup viewpoint that will be useful for future experiments at NPS.

B. PURPOSE OF EXPERIMENT 3

One of the goals of the A2C2 project is to advance the state of knowledge regarding decision-making in organizational settings. The first experiment served as a baseline and test-run of the simulation. The second experiment explored the interaction between task structure and organizational structure and drivers of adaptation in an organization (Drake, 1997). The purpose of Experiment 3 was to test research hypotheses and compare architecture performance data with pre-experiment model predictions.

This section describes the motivation behind exploring adaptation of organizational structures in an operational setting. Other sections detail the specific questions that the A2C2 research team desired to be answered and the research objectives that the Lead Team examined. The Experimental Approach section describes the architectures, the trigger event, and the schedule of runs. Other sections describe anticipated results based on model predictions, and the scope of the experiment.

1. Real-World Motivation

Futuristic warfare, like that envisioned in "Joint Vision 2010", calls for small, highly mobile, and flexible military units. The future military will be smaller and will have to react quickly to changing missions. The speed of information flow must be increased. This, in turn, will require streamlined command structures including the elimination of hierarchy that does not add value. Moreover, major military operations will be Joint, complicating command and control procedures further. With this in mind, the experiment at NPS focused on command and control architectures optimized for the assigned mission tasks.

2. Experimental Questions

The A2C2 research team objectives for this experiment were: 1) to learn more about the Joint C2 decision-making process, 2) to test the research hypothesis: organizations will adapt to an architecture closer to their current one, rather than to a more radical, though optimal, one, and 3) to compare data with model predictions. Models generated the architectures and predicted

architecture performance before and after the trigger. Also, the models calculated the "distance" between the architectures. In support of Experiment 3, the NPS Lead Team focused its analysis on the relative architecture performances (Objectives 2 and 3 above). The Lead Team research objectives were to examine:

1. The relationship between the number of decisionmaking nodes per task and architecture.
2. The average accuracy per task versus architecture.
3. The average performance per task versus architecture.
4. The possibility that learning continued throughout the experiment trials.

The measures chosen were the task accuracy as measured by the software and the task performance which was the task accuracy multiplied by the task's value. These measures will be discussed in more detail in Chapter II, Experiment Design. The first Lead Team objective was chosen to lay a foundation for the analysis of performance by analyzing architectural differences in the number of decision-makers (DMs) participating in each task. Based on observations of individuals and teams in the lab, the Lead Team also developed a method to analyze possible team learning (Lead Team Objective 4).

This thesis documents the NPS contribution to Experiment 3. The primary focus is the analysis of the Lead Team objectives in support of the A2C2 research project.

3. Experimental Approach

Guided by the A2C2 research team, the Lead Team of CC 4103 students developed and ran a joint tactical scenario, which was implemented with the Distributed Dynamic Decisionmaking-III (DDD-III) simulation software. The DDD-

III was developed by Prof. David Kleinman at UConn, and has been used previously for A2C2 experiments sponsored by the Office of Naval Research (Kleinman et al., 1996). For each of the teams, the experiment consisted of three scenario "runs". The first (pre-trigger) run was designed to be a training run and therefore was not analyzed by the A2C2 research team. Due to the nature of their research questions, in particular the possibility of learning, the Lead Team included the pre-trigger run in much of its analysis. The teams conducted the first practice run and then were presented with a "trigger". This was presented in the form of a revised Oporder and consisted of a severe reduction (~30%) in the number of assets available to the teams for the following two runs without change in mission tasks. The pre-trigger command hierarchy consisted of a six-node architecture. The distribution of assets was such that in order to accomplish tasks requiring multiple assets (mission tasks), a high degree of coordination between nodes would be necessary. Post-trigger, the players conducted the same mission tasks with reduced assets (referred to as the post-trigger setup).

The pre-trigger runs were conducted with a six-node architecture. The post trigger runs were conducted with four-, five-, and six-node architectures (see Appendix A). The post-trigger six-node architecture was the pre-trigger architecture with reduced assets. Upon notification of the asset reduction in the planning session following the first run, the teams were presented with the alternative 4- and 5-node architectures. They were then asked to choose, between the initial and the alternative architectures, the architecture they would like to use to perform the mission (see Figure 1-1). Though the trigger reduced

the total assets available to the teams, the assets in the post-trigger alternative architectures were redistributed to the remaining nodes in a manner that reduced the requirement for inter-nodal coordination relative to the pre-trigger architecture.

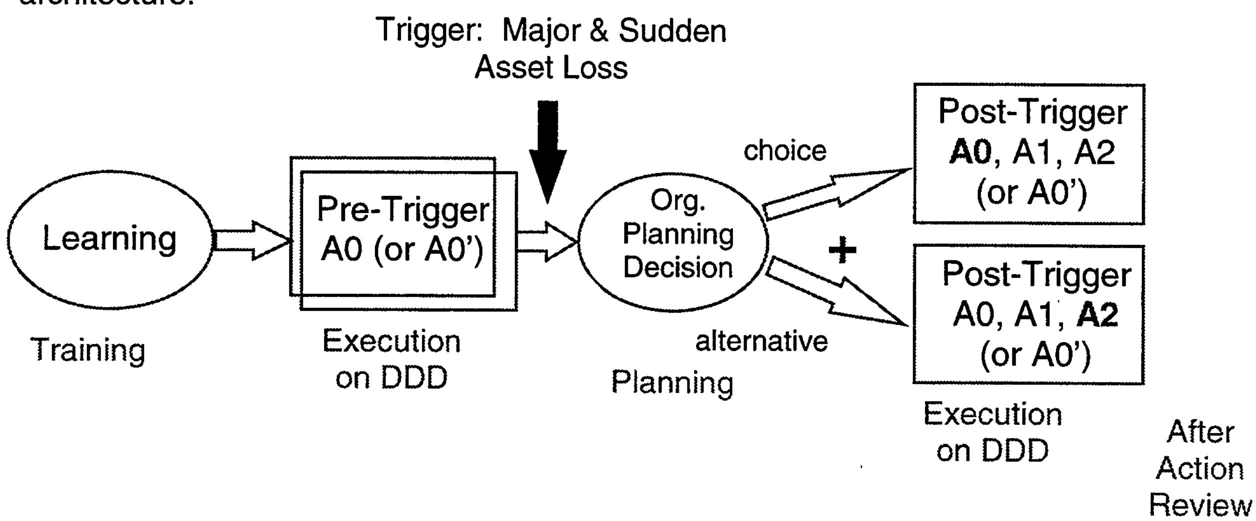


Figure 1-1: Experiment III Design (from Serfaty, 1998)

The primary mission tasks were the same for all architectures and runs: take the Hill, take the Beaches (2), take the Airport, take the Seaport, and destroy a Bridge over which counterattack forces would have to pass. Overall, there were six different asset-architecture combinations and nine "player" teams conducting three runs each for a total of 27 runs (including pre-trigger).

4. Anticipated Results

Since the Lead Team had experienced the A2C2 experiment process once before as subjects for Experiment 2 and were familiar with the DDD interface, it was believed that they could reliably comment on the predicted performance of the architectures developed for Experiment 3. Prior to the experiment, the Lead Team conducted a two-part, in-house survey. Part one

reflected how the Lead Team perceived the proximity of the post-trigger architectures to the pre-trigger A0 architecture. The average results are shown below (n=8 responses) in Figure 1-2. Zero on the scale is most similar to A0(pre) and ten is least similar.

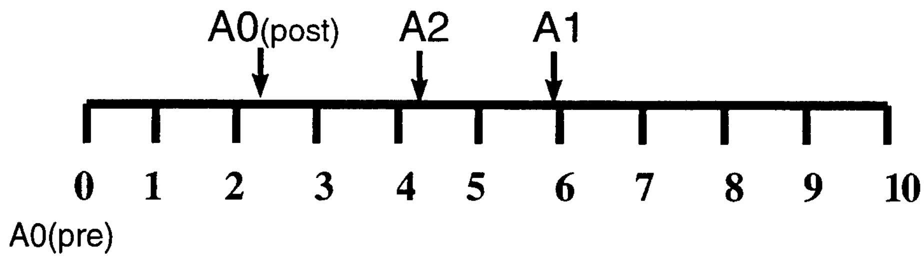


Figure 1-2: Proximity of Post-trigger Architectures Compared to A0(pre)

A model-based architecture, developed at the University of Connecticut, was designed to allocate resources, responsibility and authority in such a manner as to optimize (minimize) workload associated with inter-nodal coordination (Levchuck, et al. 1997). The architecture modelers at UConn designed the architectures with fewer nodes to require less inter-nodal coordination to successfully conduct mission tasks. Implicit in this design is the assumption that reduced inter-nodal coordination would result in better performance. The UConn modelers predicted that, based on their model, architecture A1 (4-node) would perform better than both A2 (5-node) and A0(post) (6-node). A2 was expected to perform better than A0(post) but not as well as A1. The modelers developed their predictions based on the amount of external coordination required, the geographic localization of the DMs in the architectures and the anticipated time pressure on each DM (Pattipatti and others, 1998).

Part two of the survey shows how the Lead Team expected the post-trigger architectures (A1 and A2) to perform when compared to A0(post). The average results are shown below (n=8 responses) in Figure 1-3 along with the modelers' expectations (a1 and a2) relative to A0(post). The modelers' expectations were not quantified and are displayed as a visual reference only. Zero on the scale corresponds to much worse than A0(post), while ten would be much better.

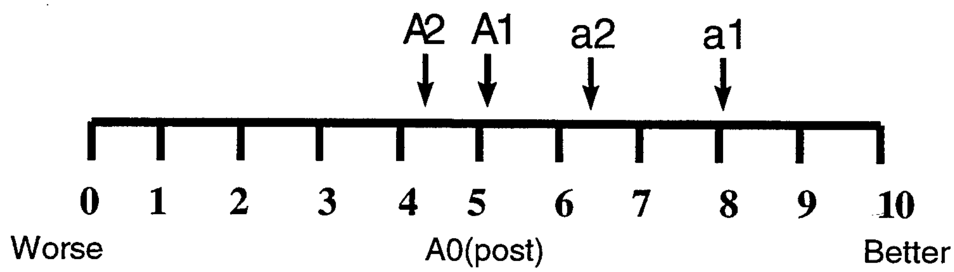


Figure 1-3: Expected Performance of Post-trigger Architectures
A1 & A2 = Lead Team avg. response, a1 & a2 = modeler

5. Scope of Experiment 3

There are many elements of human behavior that can influence the effectiveness of a military unit in a tactical situation. The A2C2 research team examined the performance of the teams under the different architectures in addition to drivers (task load, uncertainty, coordination demands, etc.) which have been shown in the past to cause teams to adapt. Post-experiment analysis will look at the performance of the teams under the different architectures and compare the results with the modeler's expected results.

The NPS Lead Team, in addition to supporting the A2C2 research team, focused data collection and analysis efforts on the relationship between inter-

nodal coordination and task performance. This analysis was scoped within the time available and the experience limits of the Lead Team.

II. EXPERIMENTAL DESIGN

A. OVERVIEW

The A2C2 experiment was developed by researchers at Alphatech, Inc., Aptima, Inc., NPS and UConn as an ONR initiative to investigate processes of adaptation in Joint C2 architectures. The specific objectives of running the third experiment were to 1) learn about Joint command and control (C2), 2) test research hypotheses, and 3) compare data with model predictions (Serfaty, 1998).

This chapter describes the details of the design of Experiment 3. A description of the physical setup of the experiment including equipment and test subjects is given. The hypothesis section includes both the A2C2 research group and the Lead Team hypotheses. Other sections describe the assumptions pertaining to the data collected, the statistical design of the experiment, the measures used by the Lead Team in their analysis, instrumentation, and pilot trial testing of the architectures.

B. SETUP

The following sections describe the laboratory environment used to support Experiment 3. The physical layout of the simulation hardware and the communications equipment are described first. Other sections detail the test subjects, special equipment used for the experiment, the schedule of the trial runs, and the specific hypotheses to be examined by the A2C2 research group in general and the Lead Team in particular.

1. Physical

The experiment was conducted in the Systems Technology Laboratory (STL) at the Naval Postgraduate School (NPS) in Monterey, California. The hardware on which the test subjects played the scenarios was a network of 6 SUN workstations running the DDD software. The workstations had partitions between them to ensure the test subjects maintained proper communications protocol and to provide some semblance of physical isolation. Communications between the test subjects were facilitated by the use of headsets at each workstation. For data collection, the communications for each run were recorded and each run was videotaped to aid the researchers in the post-experiment analysis. The workstation display was identical for each node so that a common picture was available to all team members. Individual team member units were color coded for ease of identification and coordination. Separate communications nets were used in some scenarios to simulate different command nets.

2. Test Subjects

The test subjects in the experiment were 53 military officers with varying operational experience from all services and one civilian student from both the Joint C4I and Systems Management curricula at NPS. The subjects were organized into nine six-person teams. Tactical expertise was not uniform but was evenly spread among the teams to the maximum extent possible. The scenario intentionally gave explicit mission tasks to be accomplished in a

particular order to prevent teams lacking requisite tactical skill from being unfairly handicapped during play.

3. Special Equipment

The Distributed Dynamic Decisionmaking (DDD) simulation software was the vehicle for the experiment. Prof. Kleinman (UConn) and Aptima, Inc. programmer Phil Young incorporated software modifications to the DDD (necessitated by scenario changes developed primarily at NPS) for the current experiment. Test subjects were provided with a DDD tutorial to help familiarize them with the simulation interface (see Appendix B).

Test subjects were provided with an Oporder (Appendix C), which detailed the scenario, mission, and friendly and enemy assets. Handouts were provided to the subjects before each trial run showing the mission priorities, friendly asset starting positions, and a list of tasks requiring coordination among the players. These handouts are compiled in Appendix D, and were designed to help players adjust to different command structures with minimal learning.

During the trigger planning session, a videotape made by the Lead Team was shown to the subjects to introduce them to the alternative architectures. This presentation was a simulated video-teleconference (VTC) and was designed to add a dimension of realism to the session.

Communications equipment consisted of the Sun Workstations (digital pre-formatted messages) and headsets (verbal messages) for each player. For verbal communications, the nodes could be grouped into separate communication nets to simulate "real world" communications structures.

For data collection, each planning session and trial run was videotaped and included audio input from the headsets. A separate cassette player was also used to record verbal communications during pretrial planning, the trial, and the after-action review conducted by each team following the trial run. The video- and audiotapes provide data in their own right and serve as a backup to manual survey-style data collection.

4. Schedule of Trials

The experiment at NPS took place from 12-25 November 1997. All subjects were given one hour of "buttonology" training to familiarize them with the DDD interface. Before the first trial, each team was given one hour of team training to get used to the individual station (node) they would be playing and familiarize themselves with the communications procedures. The teams were scheduled for three two-hour DDD trial runs. The first run was used for training purposes and was not designed by the researchers for data collection. The two post-trigger runs were used for data collection. As stated in the previous chapter, the Lead Team collected and analyzed data from all three runs. A two-hour planning session was conducted between the first and second runs to collect data on team decisions as a result of the internal trigger event, which severely reduced the assets available for the mission.

All teams ran a six-node architecture during their first trial (A0 or A0'). The two post-trigger trials consisted of the team's first choice as determined in the planning session, and either the A1 or A2 post-trigger alternatives. DDD post-trigger trials were counterbalanced in order to account for any learning effects

that may be present and to negate any effect due to playing the team's "choice" first (or last). In the previous chapter, Figure 1-1 depicts this design. This means that the order in which teams conducted the two post-trigger trials was mixed to the maximum extent possible. Due to time constraints, not all teams could run every architecture.

5. Hypotheses

The purpose of the third A2C2 experiment conducted at NPS was to examine the degree to which organizations will alter their structure in the face of an internal trigger event. The general statement is as follows:

> When forced to change, organizations will adapt to an architecture closer to their current structure, rather than to a more radical, though optimal, one. (Serfaty, 1998)

The expectation was that the teams would prefer to operate under a command structure (A2) that closely resembled their initial architecture rather than a more "optimal" one (A1) when presented with the trigger. Also, it was expected that, when required to do so, teams would perform better in the alternative optimal architecture.

The Lead Team looked at the relationship between coordination by players to accomplish certain tasks and the performance of these tasks. The modélers expected that team performance of these tasks would increase when fewer DMs were involved in the tasks. This would imply that the optimal architecture performed better even though it may not have been the selected

choice of the team. In support of this, analysis was done in the following four areas:

1) The number of nodes per task as a function of architecture.
2) The accuracy per task as a function of architecture.
3) The performance per task as a function of architectures.
4) The learning effect on performance.

C. ASSUMPTIONS

Various assumptions were made concerning the experiment and the data that was collected. The following sections detail these assumptions.

1. Experimental Assumptions

All teams were near the same level of understanding of how to work the DDD interface. The process of counter-balancing the runs would compensate for any learning occurring throughout the trials. Also, a team's lack of tactical expertise does not adversely affect their ability to perform on the DDD, due to the simplification of the scenario and fixed requirements to engage in tasks.

2. Statistical Assumptions

The data collected during the trials has a normal distribution. The trials run by the teams are considered independent samples of data. All data collected for a given hypothesis will have the same variance. To cover the possibility that these assumptions are invalid, Kruskal-Wallis non-parametric testing was included in the analysis.

D. STATISTICAL DESIGN OF EXPERIMENT

The analysis performed by the Lead Team was based on a single factor which, when varied, would yield valuable data for analysis. The factor chosen was the architecture, or command structure, under which each team would play. This factor was controlled at six levels corresponding to the six different architectures to be played (see Appendix A).

Three specific measures were selected for analysis. These were the accuracy of five mission tasks, the performance achieved by the teams in performing those tasks, and the number of DMs involved in performing each task. The null and the alternative hypothesis for each case can be stated as:

H_0: "The mean (μ) measure (accuracy, performance, or number of nodes per task) is the same across architectures".
H_a: "At least two of the means for each measure are different".

Or:

H_0: $\mu_1 = \mu_2 = \ldots = \mu_6$
H_a: At least 2 of the μ's are different

E. MEASURES

The specific measures collected by the Lead Team during the experiment were the accuracy achieved in performing each of the five key tasks and the number of DMs involved in each task. Accuracy was calculated by the DDD using an algorithm that compares the task requirements with the combined assets resources. Each task has a "requirements vector" listing the amount of each attribute (e.g. ground assault) required to fully complete the task and each

asset has a "resource vector" that lists the attributes that can be applied toward completing the task. Ideally, players coordinate to ensure that the combined resource vector of the assets they are combining to accomplish the task meet or exceed the task's requirements vector. The accuracy for each task and the number of DMs were extracted from the dependent variable file compiled by the DDD after each trial run. The performance on the tasks was calculated manually by multiplying the accuracy on the tasks by its individual value. Values for all tasks are given as part of Appendix E.

F. INSTRUMENTATION

The data collection instrumentation consisted of the dependent variable file compiled by the DDD at the end of each trial run. Accuracy and number of nodes participating in each task were extracted from these files for analysis on the statistical analysis package MINITAB™.

G. TESTING AND PILOT TRIALS

Upon the completion of the scenario inputs for each architecture variation, the Lead Team members play-tested each scenario to verify both the functionality of the DDD and the scenario level of difficulty. The handouts generated by the Lead Team were used and their utility checked. Minor errors were then corrected and some software changes made to the numbers of enemy assets and their movements. The pilot trials also enabled the verification of the data collection instrumentation. The dependent variable files were checked to ensure that the targeted measures were recorded and formatted correctly by the DDD.

III. DATA DESCRIPTION

A. RAW DATA

A raw data file, containing information on the tasks completed during each session, was created automatically by DDD after each run. The files contain information on the DM(s) involved in completing each task, as well as the accuracy achieved for that task. An example of a raw data file is included in Appendix F.

B. DATA CODING SCHEME

The measures described in Chapter II, Experimental Design, were automatically collected by DDD as described above. Task performance was a manual calculation involving the task accuracy from the dependent variable file and the task value. The sample raw data file in Appendix F includes annotations describing how the information is extracted from the file and what the information represents. The data coding scheme (Appendix F, part B) details how the data table (Appendix F, part C) was coded. This was necessary to perform the MINITAB™ statistical analysis.

C. DATA PROBLEMS

In several cases, a team did not complete all the specified tasks during a trial. The Lead Team performed analysis using data which included tasks that were not reached (scored as zero) as well as only those that were actually done. The difference in the analysis of tasks with zeros and the analysis of tasks excluding zeros was negligible when pertaining to the accuracy and performance

measures. However, in the analysis of team learning, significant insights were gained by both including and excluding the tasks with zeros in the analysis.

Also, the Lead Team performed analysis of the pre-trigger runs in addition to the post-trigger runs. While the A2C2 research team did not intend the pre-trigger run to be included in their analysis, the Lead Team deemed that these runs could provide useful information for some of their research questions, in particular the learning effect.

D. DATA TABLE

A condensed summary of the data collected by DDD for all trials used for this experiment is shown in table form in Appendix F.

E. DATA REDUCTION

First, pertinent data from the dependent variable files was extracted manually from the 3.5" disks. This data was then coded as described above and input into the MINITAB spreadsheet. The statistical analysis of the coded data for the measures of interest will be described in the next chapter.

IV. DATA ANALYSIS

Chapter III showed the data that the Lead Team collected and how that data was reduced prior to the analysis. This chapter shows the details of that analysis starting with the analysis plan. The methodology of the analysis is next discussed followed by the results of the testing of each hypothesis. Due to the relatively small sample sizes and the exploratory nature of their research, the Lead Team chose a probability of rejecting the null hypothesis when it is true (Type I error (α)) of 0.1 as their criterion for rejecting all hypotheses tested.

A. ANALYSIS PLAN

The Lead Team analysis plan called for the use of parametric and non-parametric analysis of variance, regression, and graphical analyses to examine the four hypotheses discussed above. This required developing measures that both could support the analysis and could be extracted from the DDD dependent variable files saved at the end of each run. These measures are discussed in Chapter III, Data Description above. MINITAB™ version 11 was selected to perform the analyses.

Since decision-maker coordination in accomplishing tasks was of primary interest, the analysis was focused on those tasks that required the successful coordination of multiple assets (e.g., CAS and INF, or 2 INF and DDG).

B. METHODOLOGY

Hypotheses one through three were first analyzed using parametric analysis of variance (ANOVA). To add strength to significant findings and robustness against violations of the normality and homogeneity of variance assumptions, Kruskal-Wallis non-parametric analysis of variance (KW) was also employed. Graphical means were used to give insights into the findings throughout the analysis.

For the fourth hypothesis (Learning), regression analysis was employed to examine the relationship between accuracy and the order in which the architectures were played. Graphical means were again used to provide added insight.

The DDD software automatically calculates an accuracy score that measures how well the assets used on a task matched the requirements to accomplish the task perfectly. For each of the complex tasks (Hill, Airport, Seaport, North Beach, and Bridge), the task accuracy score and the number of nodes used to perform the task were collected from the DDD dependent variable files. Task performance, an additional measure used in hypothesis three, was next calculated by multiplying the task accuracy score by the task value. Accuracy and performance were compared across architectures in examining hypotheses two and three respectively. These two analyses should lead to different findings only if there were an interaction between the architectures and task type as reflected in the different task weights. That is, if one architecture performed better on the low weighted tasks (Beach and Hill) and another on the

high weighted tasks (Airport and Seaport). The results do not indicate significant interaction. This is reflected in an observed correlation of .933 between accuracy and performance scores on the tasks.

C. RESULTS OF ANALYSIS

The Lead Team focused initially on three hypotheses that were chosen to answer the basic research question of whether team performance increases, on average, as the requirement for inter-nodal coordination of the architecture decreases. During the experiment, a fourth hypothesis was developed to test for the presence of a learning effect and to quantify that effect, if present. In the following paragraphs, each hypothesis is examined in turn using a combination of ANOVA, Kruskal-Wallis, linear regression, and graphical analyses. The result of each hypothesis test is given along with the computer output for the applicable analysis technique and amplifying graphical plots. A short description accompanies each graph; however, conclusions that may be drawn from the graphs are deferred until Chapter V, Conclusions.

1. Hypothesis: The number of nodes per task is the same across architectures.

It was expected prior to the experiment that the number of nodes involved in each task would be higher, based on the architecture designs and asset distribution, for the 6-node architectures. The 6-node architectures had a mix of assets that required nodes to coordinate in accomplishing tasks by matching the task requirements vector with the combined resource vector of the assets participating in the task. Appendix E contains the comprehensive list of all task

requirement vectors and asset resource vectors. The number of nodes per task was expected to decrease for the 5-node and 4-node architectures respectively since each decision-making node contained more of the assets "organically" and the asset mix was designed to reduce the external coordination requirements. This hypothesis was tested to ensure that a difference existed in the average number of nodes per task between architectures since the subsequent analysis would focus on the relationship between the number of nodes per task and task performance.

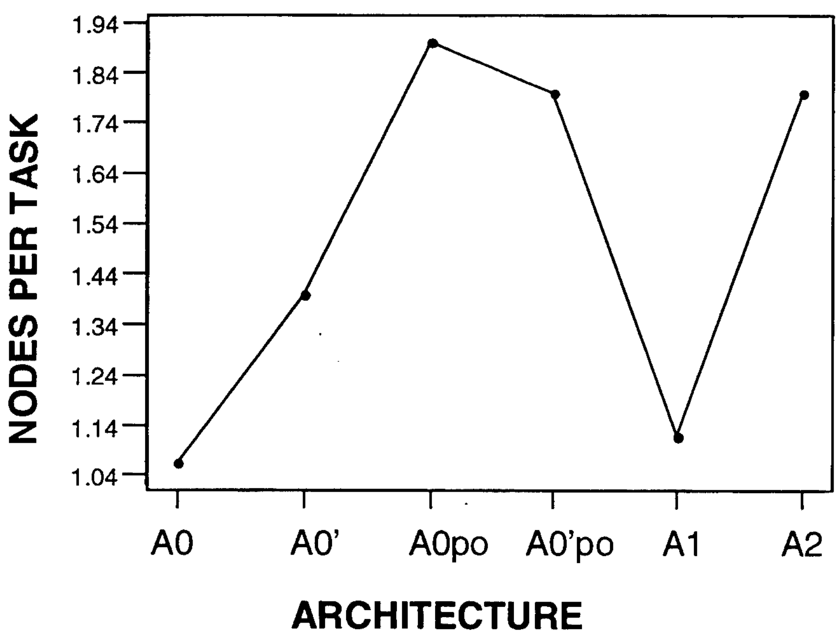


Figure 4-1: Avg. Nodes per Task By Architecture

Figure 4-1 shows the Main Effects Plot for the average number of nodes used to perform a task across the architectures. Of the post-trigger architectures, there were varying degrees of inter-nodal coordination. The A0 architectures (6-node) were expected to require the most inter-nodal

coordination, architecture A2 (5-node) would require a moderate amount of inter-nodal coordination, and A1 (4-node) would require the least. Architecture A1 was designed by the modelers to be the "optimal" architecture, requiring the least inter-nodal coordination. Each node in A1 had the majority of the assets needed to perform the tasks it would be expected to accomplish. Therefore, the nodes theoretically would not need to coordinate with each other as much as in the other architectures.

The ANOVA performed for this hypothesis showed that the number of nodes per task differs between architectures (P = 0.010). The average values in the confidence intervals are included in the MINITAB™ output below.

```
Analysis of Variance for Tsk Node
Source      DF        SS         MS         F        P
stackarc     5     17.33       3.47      3.14    0.010
Error      129    142.41       1.10
Total      134    159.73
                              Individual 95% CIs For Mean
                        Based on Pooled StDev
Level N Mean StDev -----+---------+---------+---
0    30 1.067 1.015   (------*-------)
1    15 1.400 1.183      (----------*----------)
2    30 1.900 0.995              (-------*-------)
3    15 1.800 1.014        (----------*----------)
4    25 1.120 0.927   (-------*--------)
5    20 1.800 1.240            (--------*--------)
                       ----+---------+---------+---
Pooled StDev =1.051     1.00      1.50      2.00
```

Since the ANOVA results were significant, a K-W test (see below, P = 0.017) was used for confirmation in case the assumptions did not hold. Based on these two tests, we can say with confidence that there is a difference between architectures in the number of nodes per task.

Kruskal-Wallis Test on Tsk Node

stackarc	N	Median	Ave Rank	Z
0	30	1.000	53.4	-2.32
1	15	2.000	65.3	-0.28
2	30	2.000	82.6	2.33
3	15	2.000	79.0	1.16
4	25	1.000	55.0	-1.84
5	20	2.000	78.0	1.24
Overall	135		68.0	

H = 13.73 DF = 5 P = 0.017
H = 14.70 DF = 5 P = 0.012 (adjusted for ties)

2. Hypothesis: Average accuracy is the same across all architectures.

The relationship between architecture and task accuracy was examined by the Lead Team to compare the empirical results with the modeler's predicted performance. The architectures modeled for Experiment 3 had different degrees of task-organization. That is, the decision-makers' ability to accomplish assigned tasks autonomously without coordination with other players depended on the architecture they were playing. The modelers expected the most task-organized structure to yield superior performance.

In the analysis that follows, the data are examined from several different angles. The experiment was designed with the pre-trigger runs as part of the training, and based on observation, the Lead Team suspected that a significant degree of learning was continuing to occur during pre-trigger runs. Yet, meaningful information might be available from the pre-trigger data. For these

reasons, the data were examined first by including the pre-trigger runs, and then re-examined with only the post-trigger data. This would also allow examining any differences between pre- and post-trigger results. The analysis was also conducted including and removing tasks that received zero as the accuracy score when a task was not accomplished. Removing the tasks with zeros produced similar results, which therefore are not displayed.

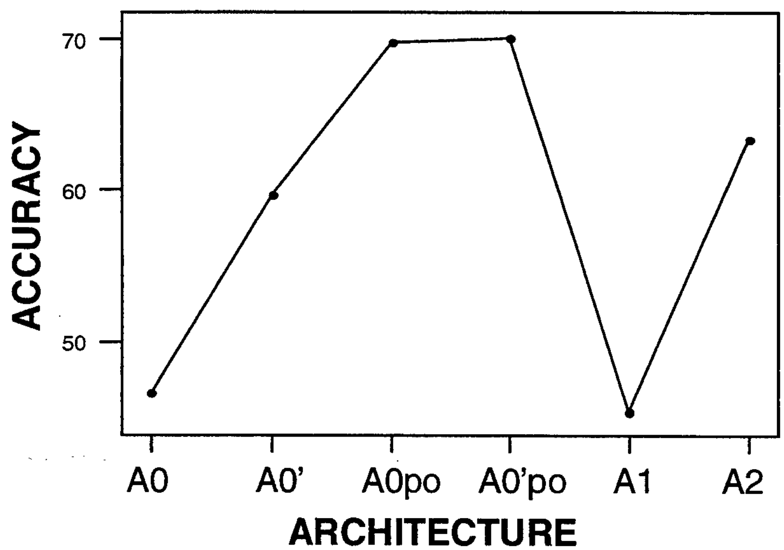


Figure 4-2: Avg. Accuracy by Architecture

The Main Effects Plot for the average accuracy of tasks across architectures was examined and the lowest instance appeared for architecture A1 as shown in Figure 4-2. So, although architecture A1 used fewer nodes per task, as shown in Figure 4-1, it also produced the lowest accuracy scores. The reasons for this will be addressed in Chapter V.

One-Way Analysis of Variance resulted in a P-value of 0.099. Therefore, we can say that architecture has an effect on task accuracy.

```
Analysis of Variance for stackacc
Source     DF        SS        MS         F        P
stackarc    5     14781      2956      1.90    0.099
Error     129    200651      1555
Total     134    215432
                                  Individual 95% CIs For Mean
                                  Based on Pooled StDev
Level       N      Mean     StDev ------+---------+---------+---------+
0          30     46.63     42.21   (------*------)
1          15     59.73     45.83       (---------*---------)
2          30     69.77     35.37              (------*------)
3          15     70.07     36.48           (---------*---------)
4          25     45.48     36.79  (-------*-------)
5          20     63.50     41.15         (--------*-------)
                                  ------+---------+---------+---------+
Pooled StDev =    39.44                 40        60        80       100
```

The analysis was again conducted with the K-W test yielding a P-value of 0.104 which agrees with the ANOVA results above.

```
             Kruskal-Wallis Test on stackacc

        stackarc     N     Median    Ave Rank         Z
        0           30      53.50        58.3     -1.55
        1           15      77.00        72.9      0.51
        2           30      81.00        77.7      1.54
        3           15      77.00        79.6      1.22
        4           25      51.00        53.9     -1.99
        5           20      86.50        73.3      0.66
             Overall    135                    68.0

             H = 8.84  DF = 5  P = 0.115
       H = 9.11  DF = 5  P = 0.104 (adjusted for ties)
```

The data is arranged below as a dotplot to more easily show the spread of the data points within each architecture.

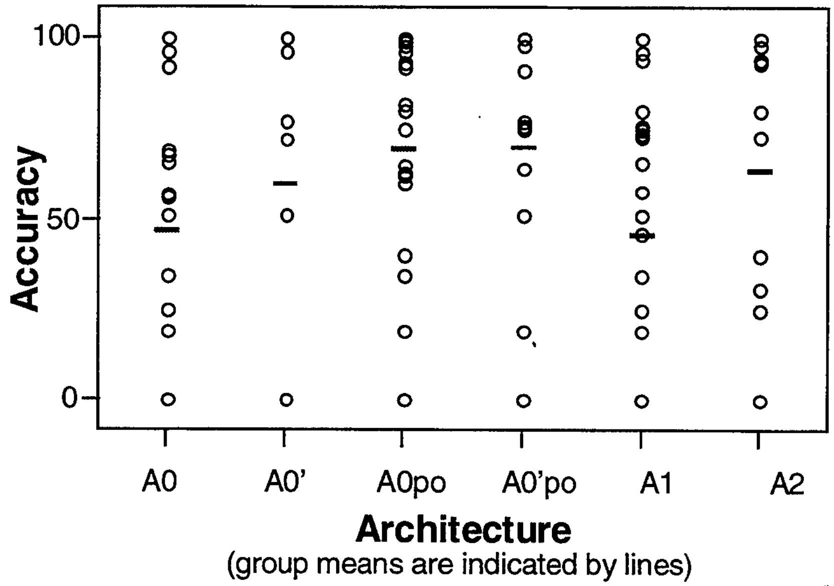


Figure 4-3: Task Accuracy by Architecture (including pre-trigger)

Figure 4-3 shows the average task accuracy across architectures. The decrease in average accuracy for architecture A1 is noteworthy and is similar to Figure 4-2.

As an excursion, the Lead Team analyzed what difference, if any, existed in the average accuracy per task as the number of nodes changed. In other words, was the average accuracy statistically the same across the number of nodes (4,5 or 6) in an architecture? The two 6-node pre-trigger runs were included in the first analysis. Then the analysis was repeated without the pre-trigger runs.

With the pre-trigger runs included, the Analysis of Variance P-value of 0.209 indicates that there is no significant difference in accuracy across the number of nodes in an architecture.

```
Analysis of Variance for nodacc
Source      DF        SS        MS        F        P
numnod       2      5055      2527     1.59    0.209
Error      132    210377      1594
Total      134    215432
                                     Individual 95% CIs For Mean
                                     Based on Pooled StDev
Level        N      Mean     StDev  -+---------+---------+---------+-----
4           25     45.48     36.79   (---------*----------)
5           20     63.50     41.15             (----------*-----------)
6           90     60.43     40.46                 (----*-----)
                                    -+---------+---------+---------+-----
Pooled StDev =     39.92            30        45        60        75
```

Figure 4-4 below confirms the lack of significant differences in average task accuracy when there was a different number of nodes in the architecture. There are many more data points associated with 6 nodes since both pre-trial architectures consisted of 6 nodes. Architecture A1 had 4 nodes, and A2 had 5 nodes.

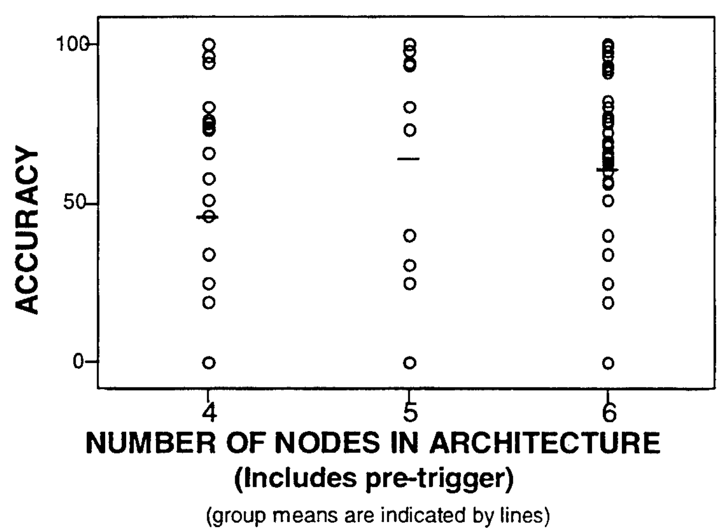


Figure 4-4: Task Accuracy by Number of Nodes in Architecture

The data was then re-examined without the pre-trigger runs in order to see if the results would change. Analysis of Variance resulted in a P-value of 0.034

which indicates that there is a significant difference in accuracy scores when only the post-trigger runs are included. The difference between these results and those generated including all runs will be discussed in Chapter V, Conclusions.

```
Analysis of Variance for 2stacacc
Source     DF        SS        MS        F        P
staknumn    2      9643      4822     3.51    0.034
Error      87    119578      1374
Total      89    129222
                                   Individual 95% CIs For Mean
                                   Based on Pooled StDev
Level       N      Mean     StDev  ----------+---------+---------+------
4          25     45.48     36.79  (---------*---------)
5          20     63.50     41.15             (----------*----------)
6          45     69.87     35.33                    (-------*------)
                                   ----------+---------+---------+------
Pooled StDev =    37.07                     45        60        75
```

The output of the K-W test using this data is shown below. The P-value of 0.024 confirms the significant difference between accuracy scores.

```
Kruskal-Wallis Test on 2stacacc

staknumn        N    Median    Ave Rank         Z
4              25     51.00        33.5     -2.69
5              20     86.50        47.8      0.44
6              45     80.00        51.1      2.05
     Overall     90                     45.5

        H = 7.49   DF = 2   P = 0.024
H = 7.63   DF = 2   P = 0.022 (adjusted for ties)
```

Figure 4-5 is a dotplot of this data and illustrates the significant differences in average task accuracy. With the two pre-trigger runs removed, the results indicate that the more nodes in the architecture, the better the resulting task accuracy.

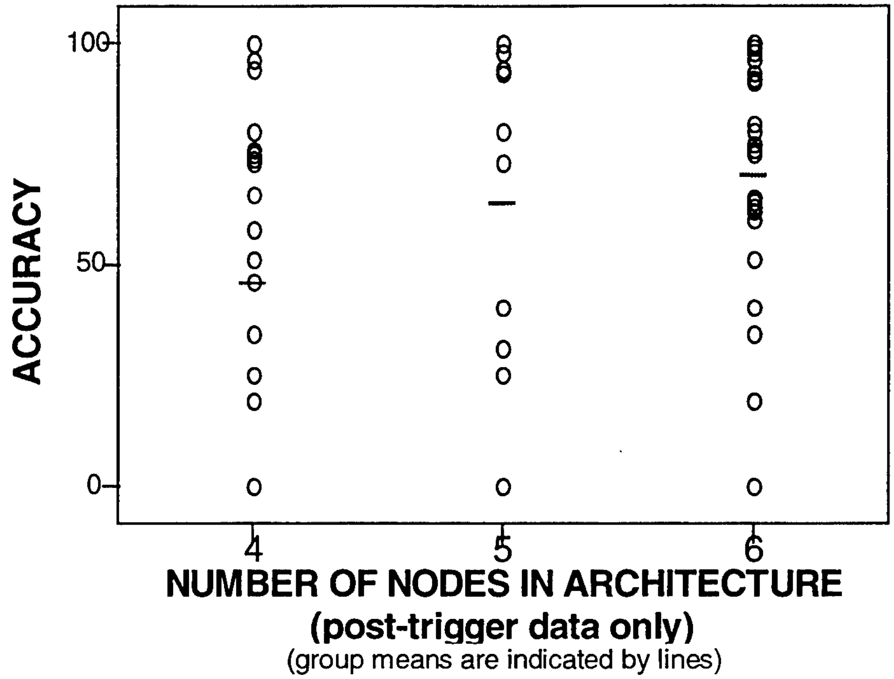


Figure 4-5: Task Accuracy by Number of Nodes in Architecture

Another excursion related to hypothesis two examines the relationship between average accuracy and the number of nodes used in each individual task. A sub-hypothesis is stated as follows: Average accuracy is the same across the number of nodes used on a task. If the subjects did not accomplish a task, both the accuracy score and the number of nodes used for the task were assigned a value of zero. The Analysis of Variance P-value of 0.000 indicates a highly significant difference in accuracy scores across the number of nodes used. This result is validated by the K-W test below.

```
Analysis of Variance for stackacc
Source     DF        SS        MS         F        P
Tsk Node    4    189313     47328    235.57    0.000
Error     130     26119       201
Total     134    215432
                                   Individual 95% CIs For Mean
                                   Based on Pooled StDev
Level       N      Mean     StDev  --+---------+---------+---------+----
0          33     -0.00      0.00   (*)
1          32     48.00     22.71             (*)
2          42     85.00     15.20                     (*)
3          27     97.78      5.05                        (*-)
4           1    100.00      0.00                   (------*------)
                                   --+---------+---------+---------+----
Pooled StDev =    14.17               0        40        80       120
```

```
              Kruskal-Wallis Test on stackacc

         Tsk Node     N    Median    Ave Rank         Z
         0           33  0.00E+00        17.0     -8.62
         1           32  5.10E+01        55.1     -2.14
         2           42  9.20E+01        89.8      4.34
         3           27  1.00E+02       109.9      6.22
         4            1  1.00E+02       119.0      1.31
             Overall    135                   68.0

              H = 105.28  DF = 4  P = 0.000
         H = 108.48  DF = 4  P = 0.000 (adjusted for ties)
```

Note from the above ANOVA that the accuracy scores are different even if the zero tasks are excluded from the analysis.

Figure 4-6 below shows the results of the dotplot between accuracy and number of nodes used on a task. A positive slope exists and suggests that tasks on which a higher number of nodes were used were accomplished with a higher level of accuracy. Thus, coordinated attacks between multiple decision-makers produced more desirable results in the form of higher accuracy per task.

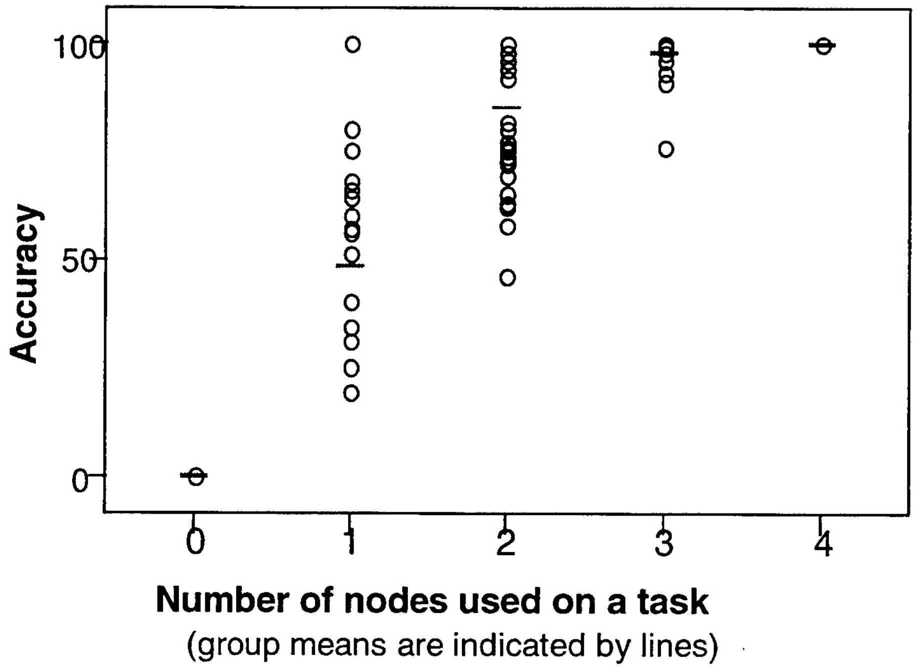


Figure 4-6: Task Accuracy by Number of Nodes Used

3. Hypothesis: Average Performance is the same across all architectures.

The third hypothesis is similar to the second. The difference is that now the individual weighting of each task as assigned by the NPS researchers is included in the analysis. Results of the performance measure are closely correlated to the accuracy measure in hypothesis 2. Each task had an associated value (see Appendix E) which was unknown to the subjects. This value was multiplied by the team's accuracy in performing that task and then added to the team score as it was accomplished. The values for the tasks included in the Lead Team analysis were 30 each for the Airfield and Seaport, 10 each for the Hill and Beach, and 15 for the Bridge. The performance was

calculated by multiplying the value of the individual task with the average accuracy associated with that task.

As with the second hypothesis, the data was subdivided to explore the differences between analyses that include and exclude pre-trigger runs. The results presented below include tasks that received zeros as performance scores. Removing the tasks with zeros produced similar results, which therefore are not displayed.

The Analysis of Variance P-value of 0.088 indicates that architecture has an effect on task performance.

```
Analysis of Variance for task per
Source     DF        SS        MS        F        P
perf arc    5   9951202   1990240     1.97    0.088
Error     129 130454288   1011274
Total     134 140405490
                                   Individual 95% CIs For Mean
                                   Based on Pooled StDev
Level       N      Mean     StDev  -+---------+---------+---------+-----
0          30       776       928   (--------*--------)
1          15       877       865  (------------*------------)
2          30      1422      1104                  (---------*--------)
3          15      1260      1021           (-----------*------------)
4          25       793       876   (---------*---------)
5          20      1249      1185            (----------*----------)
                                   -+---------+---------+---------+-----
Pooled StDev =     1006           400       800      1200      1600
```

The significance of the ANOVA is echoed by the following K-W test. The P-value of 0.099 suggests that performance was not constant across all architectures.

Kruskal-Wallis Test on task per

perf arc	N	Median	Ave Rank	Z
0	30	565.0	57.2	-1.71
1	15	960.0	65.2	-0.30
2	30	1000.0	81.3	2.12
3	15	1000.0	78.5	1.11
4	25	580.0	56.8	-1.59
5	20	865.0	72.4	0.55
Overall	135		68.0	

H = 9.25 DF = 5 P = 0.099
H = 9.42 DF = 5 P = 0.093 (adjusted for ties)

Figure 4-7 shows the average task performance by architecture. The lower value for architecture A1 is consistent with previous analyses.

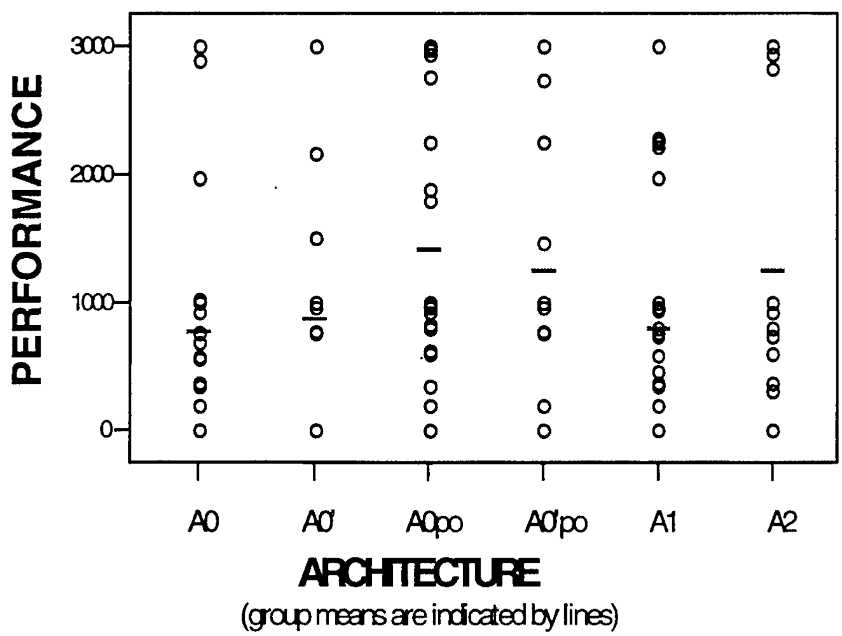


Figure 4-7: Task Performance by Architectures

As an excursion, the relationship between average performance and the number of nodes in the architecture was examined. The two 6-node pre-trigger runs were included in the first analysis. Then the analysis was repeated without the pre-trigger runs. The Analysis of Variance P-value of 0.182 is not significant and thus the hypothesis that average performance is the same across the number of nodes in the architectures is not rejected. This is similar to hypothesis 2 results, which examined average accuracy. There appears to be little difference between average accuracy and performance results when the pre-trigger runs are included.

```
Analysis of Variance for nodperf
Source      DF        SS        MS         F        P
numnod       2   3505875   1752938      1.73    0.182
Error      132 134117507   1016042
Total      134 137623382
                                   Individual 95% CIs For Mean
                                   Based on Pooled StDev
Level        N      Mean     StDev ---+---------+---------+---------+---
4           25       708       831  (---------*---------)
5           20      1249      1185            (----------*----------)
6           90      1039      1011             (----*----)
                                   ---+---------+---------+---------+---
Pooled StDev =      1008            400       800      1200      1600
```

Figure 4-8 illustrates that average task performance is not significantly different across the number of nodes in an architecture when pre-trigger runs are included.

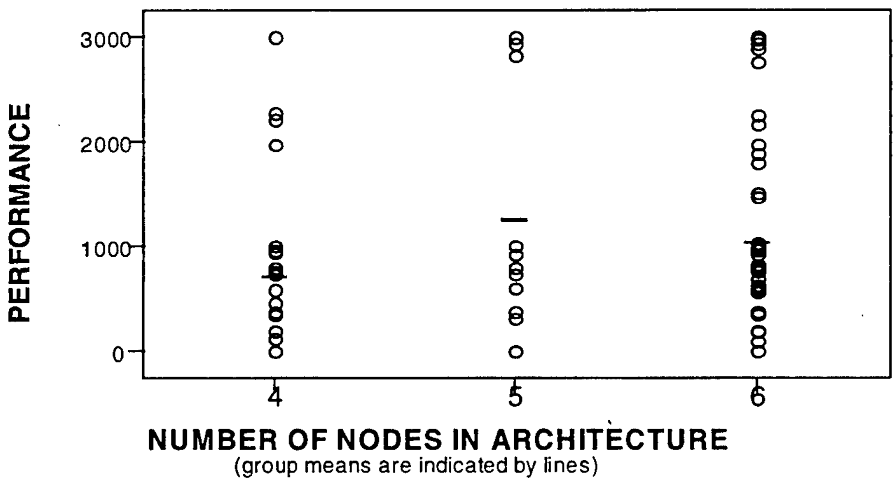


Figure 4-8: Task Performance by Number of Nodes in Architecture

The same analysis was then performed without the pre-trigger runs. The Analysis of Variance P-value of 0.060 indicates that there is a difference in average performance across the number of nodes in the architecture. When only the post-trigger runs are examined, the number of nodes in the architecture has an effect on team performance.

```
Analysis of Variance for 2stacper
Source     DF        SS        MS        F        P
staknumn    2   6216536   3108268     2.91    0.060
Error      87  92851453   1067258
Total      89  99067989
                                    Individual 95% CIs For Mean
                                    Based on Pooled StDev
Level       N      Mean     StDev   ---+---------+---------+---------+---
4          25       708       831   (----------*---------)
5          20      1249      1185                (-----------*-----------)
6          45      1311      1062                     (-------*------)
                                    ---+---------+---------+---------+---
Pooled StDev =     1033               400       800      1200      1600
```

The dotplot for this data shown in Figure 4-9 below illustrates the differences in average task performance when the two pre-trial runs with 6 node architectures were removed. This data suggests that architectures with more nodes yield better task performance.

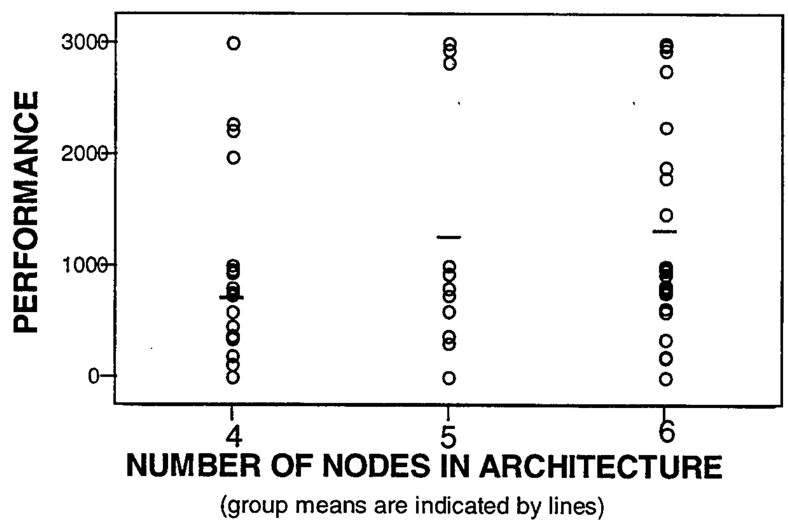


Figure 4-9: Task Performance by Number of Nodes in Architecture

Another excursion related to hypothesis three examines the relationship between average performance and the number of nodes used in each individual task. A sub-hypothesis is stated as follows: Average performance is the same across the number of nodes used on a task. If the subjects did not accomplish a task, both the performance score and the number of nodes used for the task were assigned a value of zero. The Analysis of Variance P-value of 0.000 indicates a significant difference in performance scores across the number of nodes used. This result is validated by the K-W test below.

```
Analysis of Variance for 2stacper
Source      DF        SS        MS        F        P
Tsk Node     4  64615822  16153956    27.71    0.000
Error      130  75789668    582997
Total      134 140405490
                                     Individual 95% CIs For Mean
                                     Based on Pooled StDev
Level        N      Mean     StDev   --+---------+---------+---------+----
0           33      -0.0       0.0   (-*-)
1           32     857.2     720.5         (-*)
2           42    1588.5     943.7              (-*)
3           27    1690.4     944.3              (-*-)
4            1    3000.0       0.0               (---------*---------)
                                     --+---------+---------+---------+----
Pooled StDev =     763.5               0      1500      3000      4500
```

```
            Kruskal-Wallis Test on 2stacper

Tsk Node      N     Median   Ave Rank         Z
0            33   0.00E+00       17.0     -8.62
1            32   6.00E+02       63.3     -0.77
2            42   1.00E+03       90.0      4.40
3            27   1.00E+03       99.3      4.65
4             1   3.00E+03      129.5      1.58
     Overall    135                  68.0

            H = 89.67  DF = 4  P = 0.000
     H = 91.39  DF = 4  P = 0.000 (adjusted for ties)
```

Note from the above ANOVA that the accuracy scores are different even if the zero tasks are excluded from the analysis.

Figure 4-10 below shows the dotplot between performance and number of nodes used on a task. A positive slope exists and suggests that tasks on which a higher number of nodes were used were accomplished with a higher level of performance. Thus, coordinated attacks between multiple decision-makers produced more desirable results in the form of higher performance per task. In other words, if more decision-makers contributed assets on a given task, a higher performance resulted.

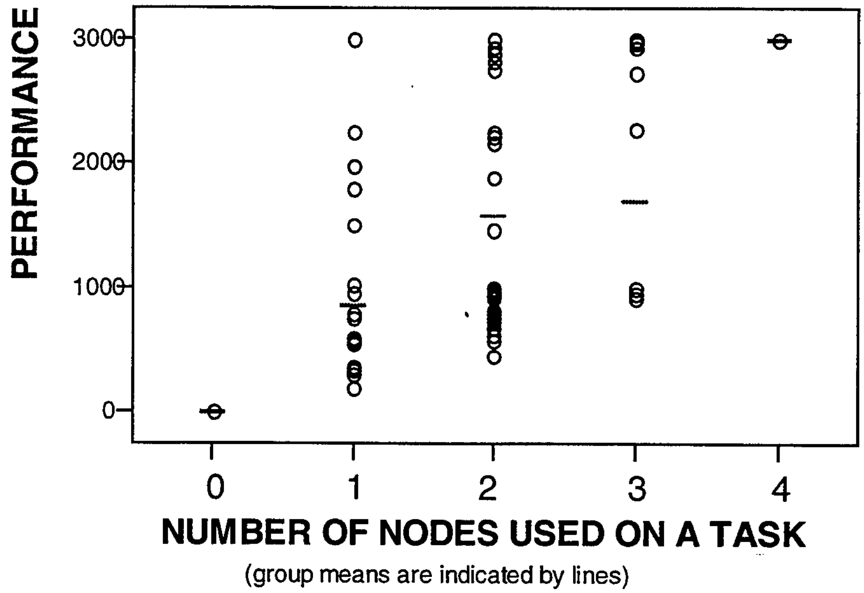


Figure 4-10: Task Performance by Number of Nodes used on a Task

4. Hypothesis: Accuracy per task is the same across the order of trial runs.

The final hypothesis examined by the Lead Team was derived from the possibility that subject teams continued to improve performance throughout the sequence of runs. The basic assumption that each subject team had a solid understanding of the DDD interface and how to accomplish the assigned tasks required of each decision-maker was a topic of particular interest to the Lead Team. It was hoped to determine whether teams continued to improve (learn) as they completed more runs.

Based on the Analysis of Variance P-value of 0.134 we cannot reject the hypothesis that accuracy was not affected by the order of the trial runs.

```
Analysis of Variance for stackacc
Source      DF        SS        MS        F        P
stackord     2      6460      3230     2.04    0.134
Error      132    208972      1583
Total      134    215432
                                   Individual 95% CIs For Mean
                                   Based on Pooled StDev
Level        N      Mean     StDev  --------+---------+---------+--------
0           45     51.00     43.38   (--------*---------)
1           45     55.87     41.90       (---------*--------)
2           45     67.49     33.35                (---------*---------)
                                    --------+---------+---------+--------
Pooled StDev =     39.79                   48        60        72
```

The dotplot below (Figure 4-11) illustrates this relationship. Although this figure shows a slight upward trend in accuracy scores as more trial runs are conducted, the ANOVA results do not conclusively support the theory that accuracy improved as runs were completed.

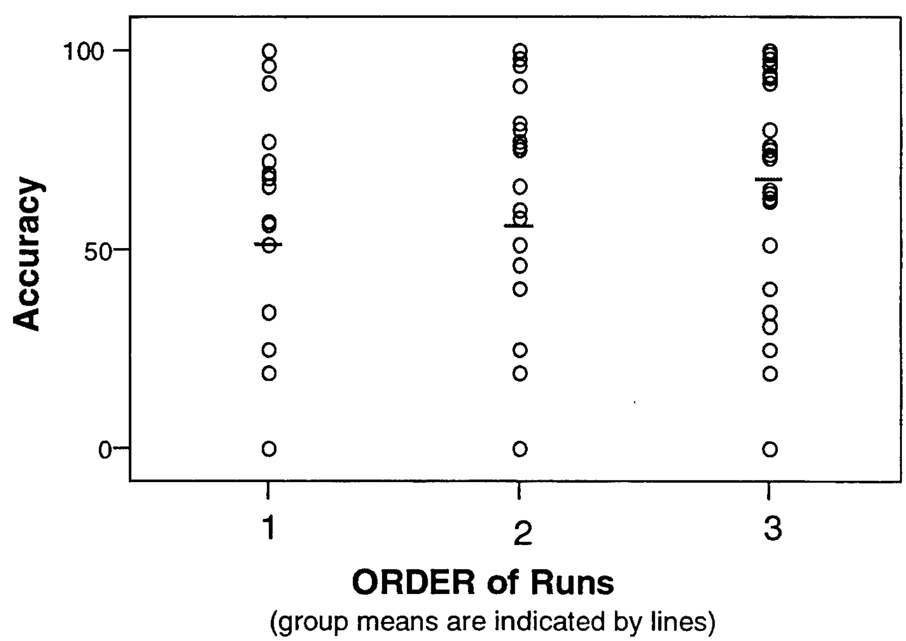


Figure 4-11: Task Accuracy by Order of Runs

Linear regression was next used to see if the observed positive slope could have been simply the result of chance. The results are displayed below in Figure 4-12. The first trial run for all teams was pre-trigger and, since it was not intended as a data collection run, it resulted in an inordinate number of tasks that were not performed. This was because the teams were not sufficiently familiar with the interface and mission requirements. These tasks that received a zero were initially included in the regression analysis. This may have contributed to the positive slope in the graph since these data points would cause the average accuracy for the first run to be lower than the post-trigger runs and tended to "pull down" the left half of the line.

```
The regression equation is
stackacc = 49.9 + 8.24 stackorder

Predictor        Coef        StDev          T        P
Constant       49.874        5.399       9.24    0.000
stackord        8.244        4.182       1.97    0.051

S = 39.67       R-Sq = 2.8%       R-Sq(adj) = 2.1%

Analysis of Variance

Source        DF          SS        MS         F        P
Regression     1        6117      6117      3.89    0.051
Error        133      209315      1574
Total        134      215432
```

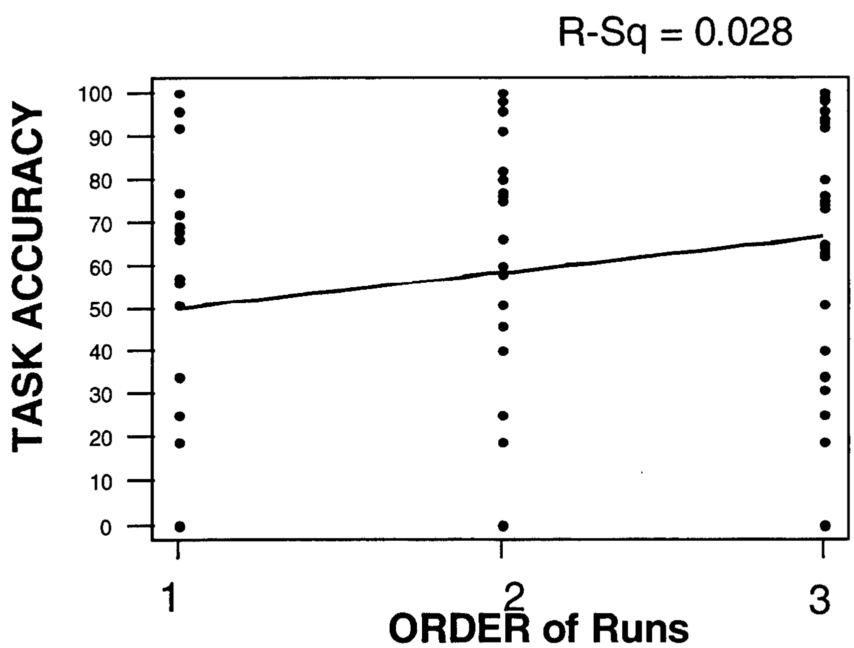


Figure 4-12: Regression Plot of Task Accuracy by Order of Runs

Now contrast these results with the output obtained by dropping those tasks that were not performed at all. These were tasks for which the software recorded a score of zero and were not actually attempted by the teams.

```
The regression equation is
nzacc = 78.6 - 1.54 nzorder

Predictor        Coef        StDev           T         P
Constant       78.623        4.348       18.08     0.000
nzorder        -1.536        3.161       -0.49     0.628

S = 26.03        R-Sq = 0.2%       R-Sq(adj) = 0.0%

Analysis of Variance

Source       DF         SS            MS          F         P
Regression    1      159.9         159.9       0.24     0.628
Error       100    67743.5         677.4
Total       101    67903.4
```

The plot of the accuracy versus the order of runs for data excluding zeros is displayed below as Figure 4-13. Without the zero tasks dragging down the average accuracy for the first run, the line is nearly level.

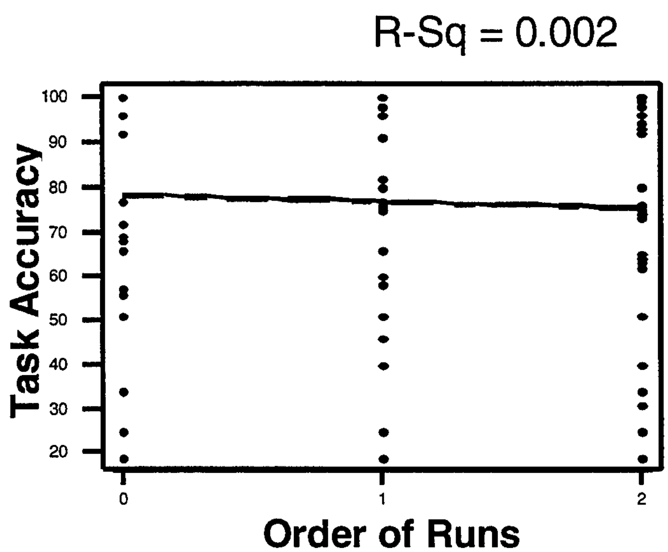


Figure 4-13: Regression Plot of Task Accuracy by Order of Runs

Another regression test was done to see if the number of tasks completed increased decisively as teams completed more runs. Figure 4-14 below shows a regression test of the number of tasks that were completed by the number of trial runs conducted. There is a slight indication that the number of tasks completed increases as more trial runs are conducted. This could indicate learning on behalf of the participants.

```
The regression equation is
numtasks = 3.19 + 0.556 Order

Predictor        Coef        StDev          T        P
Constant       3.1852       0.2406      13.24    0.000
Order          0.5556       0.1864       2.98    0.006

S = 0.7907      R-Sq = 26.2%     R-Sq(adj) = 23.3%

Analysis of Variance

Source        DF        SS          MS          F        P
Regression     1      5.5556      5.5556      8.89    0.006
Error         25     15.6296      0.6252
Total         26     21.1852
```

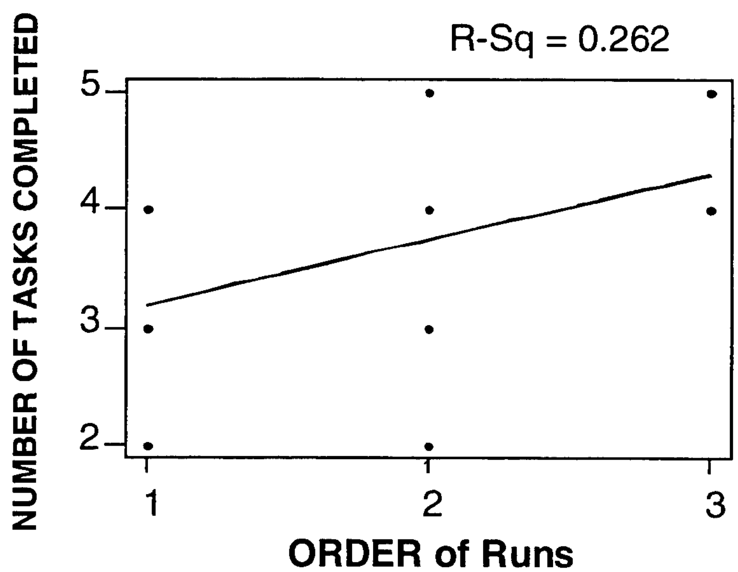


Figure 4-14: Regression Plot of Number of Tasks Completed by Order of Runs

V. CONCLUSIONS

Due to the relatively small sample sizes and the exploratory nature of their research, the Lead Team chose a probability of Type I error (α) of 0.1 as their criterion for rejecting all hypotheses tested. Besides presenting the reject/fail to reject results, P-values are also included to report the actual significance observed.

A. HYPOTHESIS RESULTS - INTERPRETATIONS

1. Hypothesis Number One

The first hypothesis that the Lead Team examined compared the average number of nodes participating in a task across architectures. A central belief of the modelers was that the inter-nodal task coordination requirements would decrease as the architectures became more task-organized. An indicator of this theory would be a result showing the greatest number of nodes participating in tasks for the A0 architectures, fewer nodes participating in tasks for the A2 architecture, and the smallest number of nodes participating in tasks for the A1 architecture.

The first run of all teams was in either the A0 or A0' architecture. Post trigger, the teams ran either the A0, A0', A2 or A1 architecture. The A0 architecture was designed to resemble a more traditional structure where assets were spread along generic, functionally organized warfighting roles such as Sea Defense, Amphibious Assault, and Air Warfare. Both the pre-trigger and post-trigger A0 architectures spread the assets across six DMs in such a way that no

single decision-maker possessed the necessary assets to sate all of the requirements vectors of complex tasks. The other architectures were designed so that the level of task-organization increased for the post-trigger A2 and A1 architectures respectively, increasing the proportion of times the DMs were provided the necessary assets to perform their node's respective tasks. The A2 (5-node) architecture was somewhat more task-organized, and the A1 (4-node) architecture showed the most task organization. Therefore, the number of nodes participating in each task should be relatively high for the A0 and A0' architectures (6-node) and lowest for the more task-organized A1 architecture (4-node), with the A2 architecture (5-node) somewhere in the middle.

The Main Effects Plot shown in Figure 4-1 illustrates the empirical results. For the post-trigger runs, the A0 and A0' architectures show the highest average number of nodes per task, and the A1 architecture shows the lowest average nodes per task. This was expected and agrees with the modelers' prediction. An interesting result was the relatively low number of nodes per task of the pre-trigger architectures. As stated above, the pre-trigger run was intended to be a training exercise and the A2C2 researchers did not include it in their analysis. It is likely that the teams were continuing to familiarize themselves with the DDD interface which led to the development of the fourth hypothesis that the Lead Team examined. Based on the ANOVA and K-W tests, we conclude that the average number of nodes per task is different between architectures ($P = 0.01$). The hypothesis that the average nodes per task is the same for each architecture is therefore rejected.

2. Hypothesis Number Two

The second hypothesis examined whether the average accuracy score per task changed between architectures. The architecture modelers expected that the accuracy of tasks would be the highest for architectures requiring the least inter-nodal coordination. The Main Effects Plot in Figure 4-2 does not support this theory. The plot shows that of the post-trigger architectures, A1, which involved the lowest inter-nodal coordination, produced the lowest average accuracy. So, even though the DMs possessed most of the assets required to accomplish complex tasks on their own, they did not bring all of the resources required to bear on the task. With only four DMs in architecture A1, it is quite possible that task saturation occurred, and the DMs may have made insufficient attacks in order to complete all of the mission tasks.

Based on the ANOVA and K-W P-values of .099 and 0.115 respectively, the hypothesis that the average accuracy score was the same across all architectures is rejected for the post-trigger architectures. However, based on Figure 4.2, it does not differ between the post-trigger A0 and A0' architectures which both had six nodes.

An excursion hypothesis that the average accuracy per task differed as the number of nodes (4,5,or 6) changed could not be verified when both pre- and post-trigger runs were considered. The ANOVA produced a P-value of .209, which is considered inconclusive. The dotplot in Figure 4-4 also indicated no significant difference. Next, to isolate the influence of pre-trigger runs that possibly reflected team learning, post-trigger only data was analyzed. With the

pre-trigger runs dropped from the analysis, the ANOVA yielded a P-value of .034, which is considered significant. This disparity is caused by the pre-trigger runs, which were all 6-node architecture-based and contained a large number of runs with low accuracy as illustrated by the Main Effects Plot in Figure 4-2. If we discount these pre-trigger runs, it can be said that the 5 and 6-node architectures yielded greater average accuracy per task than the 4-node architecture (see Figure 4-5).

As another excursion under this hypothesis, the Lead Team analyzed the relationship between the number of nodes per task and accuracy. The first step was to determine whether the accuracy varied with the nodes per task. The ANOVA and K-W P-values of 0.000 validate the theory that accuracy varied with the number of nodes involved in a task. The next step was to determine the form of the relationship. Figure 4-6 shows that accuracy and number of nodes used are positively correlated; the more DMs coordinating to accomplish a task, the higher the accuracy. For the most part, nodes that performed tasks autonomously failed to correctly match the asset's resource vector to the task's requirement vector. During the runs, the Lead Team visually verified mismatched attacks and this analysis backs up their observations. When more players participated on coordinated multi-asset attacks, more care was taken to bring the required assets to bear. A likely reason for this was inadequate pre-experiment training of the subjects in the way single node, multiple asset attacks are performed.

In summary, hypothesis two is rejected. The post-trigger analysis revealed that task accuracy differed between the architectures that had different numbers of nodes. The accuracy was significantly different for the 4,5, and post-trigger 6-node architectures.

3. Hypothesis Number Three

Analysis of the third hypothesis nearly mirrored the analysis performed in hypothesis two. The difference was the use of performance instead of accuracy as the measure. The ANOVA P-value of .088 is consistent with previous analysis above on accuracy (P-value of .099).

As with accuracy, the effect of the number of nodes on performance was examined. Similar results were concluded from analysis including and excluding the pre-trigger runs. Again, the results are only significant when the pre-trigger runs are excluded (P-value = .060). For the post-trigger runs, performance differed as the number of nodes in the architecture varied.

The test on whether the number of nodes (excluding pre-trigger runs) participating in an attack affects team performance showed a similar upward trend as shown for accuracy in the previous hypothesis (Figure 4-10). As the number of DMs involved in a task increased, the performance of the team increased.

As discussed in Chapter IV, analyses using accuracy and performance as the measures should lead to different conclusions only if an interaction existed between the architectures and task type as reflected in the different task weights. That is, if one architecture performed better on the low weighted tasks (Beach

and Hill) and another on the high weighted tasks (Airport and Seaport). The results do not indicate significant interaction. This is confirmed by a correlation between accuracy and performance scores on the tasks of .933. They are essentially measuring the same thing.

The findings contradict the pre-experiment expectation that the highest performance should result from the architecture that required the least inter-nodal coordination in accomplishing tasks.

Why is there such a disparity in the predicted architecture performance and actual results? One reason for the poor performance of the 4-node architecture may be that the subjects were inadequately trained in performing autonomous complex mission tasks. Since the majority of the training of the subjects occurred under the 6-node architecture, they were more comfortable when coordinating with other nodes to perform tasks. Another possible reason for poorer than expected performance was time pressure. It could be that teams, or individual nodes, recognized during the first run that time was a factor and that they had to increase their pace to ensure that all mission tasks were accomplished. This is confirmed by the steady increase in the number of tasks performed as teams did more runs. Still another reason for poor performance may have been task saturation. In the post-trigger runs, players may have become overloaded and may have consciously conducted less than optimal attacks in order to maintain the tempo of the experiment and complete all of the mission tasks. This would result in some attacks with very low accuracy scores

and an overall increase in the number of tasks accomplished as more runs were completed.

The analysis, along with Lead Team observations, suggests that the simulation interface played a significant part in the player team's perceptions of which architectures were more desirable. In the planning sessions, numerous statements were made by the players alluding to the difficulties encountered in performing both the one-node tasks and complex coordinated attacks. As stated above, the training of the subjects in making proper attacks may have been insufficient and subsequently reflected in team performance. This lack of proficiency seemed to play a part in all teams opting to remain in a 6-node architecture. The interface requirements, when performing multi-asset attacks by one node, seemed to affect the player's willingness to attempt these tasks. Even when players possessed the assets required to autonomously accomplish tasks, they sought peer's help. Unfamiliarity with the other post-trigger architectures and their correspondingly different asset mixes may have caused some players to decide to play under the "known" architecture. Having a higher degree of confidence with the pre-trigger structure and the assets under their control may have also resulted in the post-trigger 6-node architectures receiving the highest performance scores. This apparent lack of proficiency led the Lead Team to pursue the question of whether learning affected team performance.

4. Hypothesis Number Four

The final Lead Team research question focused on the concept of team learning. The Lead Team decided to determine whether the accuracy per task

changed as the teams did more runs. The order of trial runs versus accuracy was examined using ANOVA and resulted in a P-value of 0.134. These results are not strong enough to reject the hypothesis that the accuracy per task is the same across the order of the trial runs.

Since ANOVA does not test for the presence of trends, linear regression was also used. The linear regression analysis produced some interesting results. The analysis was complicated by the presence of tasks that were not accomplished due either to the scenario time limit being reached or omission. When they were included, the incomplete tasks received a score of zero. Analysis was performed both with and without these tasks. First, regression was done on the accuracy versus the order in which teams did runs. When the tasks that received a zero were included, the regression resulted in a slight but significant upward slope ($P = .051$). However, when the tasks receiving a score of zero were omitted from the analysis, the regression line actually took on a negative, though insignificant slope ($P = .628$). Finally, it was determined that the number of tasks completed increased as the teams did more runs ($P=.006$) which reflects a team becoming more familiar with the mission tasks.

This analysis suggests that learning was occurring throughout. When the tasks not completed were assigned an accuracy of zero and were included in the analysis, the accuracy increased from run to run, and when they were omitted, the number of tasks accomplished increased from run to run without loss of accuracy. This is consistent with the belief of the Lead Team observers that player proficiency was lacking into the post-trigger runs. In some post-

experiment responses, players stated that their performance increased as they did more runs and became more familiar with the interface and scenario. Adequate training should, in the future, reduce this effect to an insignificant level.

B. EXPERIMENT SUMMARY

The analysis process is complex. Choosing Measures of Performance, deciding on a manner to collect data for these measures, and analyzing and interpreting the data was a continuous learning process for the author and the Lead Team. The analysis produced significant results in support of the four Lead Team hypotheses chosen. The two measures of accuracy and performance were closely correlated meaning that the different architectures did no better or worse on tasks that had a high task value associated with them. Had certain architectures done better on high value tasks at the expense of low value mission tasks, the correlation would not have been as high.

The learning effect must be addressed in future experimentation. Experiment objectives and scope must take into account pre-experiment requirements regarding training to avoid confounding the results.

VI. LEAD TEAM LESSONS LEARNED AND AREAS FOR FUTURE EXPERIMENTATION

This chapter will examine some of the key responsibilities of the Lead Team that if not adequately addressed can have an adverse impact on the conduct of experimentation at NPS. This chapter is divided into three parts. First, lessons learned by the Lead Team pertaining to the experiment preparation phase are examined. Second, the lessons from the experimentation phase are discussed. Finally, areas and issues for future experimentation in support of A2C2 are presented.

A. EXPERIMENT PREPARATION

In addition to their involvement in the design, data collection, and analysis of measures as part of a class project, the Lead Team provided key support for Experiment 3 by performing a variety of administrative functions both before and during the experiment. This involvement began at the start of the quarter in which the experiment was run and required many concurrent tasks to be accomplished. The following discuss the areas the author believes should be of greatest concern to anyone involved in future experimentation at NPS, particularly to the lead member of the Lead Team.

1. Scheduling

The first major task in preparing for the experiment is scheduling. This includes deconfliction of the Lead Team and the players' schedules. Also, if a requirement exists for Lead Team observers to be present at each run, a

feasibility analysis should be done early in the quarter so that if a problem with a certain time slot occurs, observer substitutes can be arranged. Close interaction is required in the scheduling process with those instructors providing players to the experiment. Unless the senior C4I class has a clearly defined task delegation schedule showing which Lead Team member is responsible for what and when, one or two people will end up doing all the work. Sufficient time should be built into the tasking to allow for minor modifications as the experiment draws closer.

2. Simulation Modification

Prior to the actual run of the experiment, potential pitfalls exist that, if not promptly planned for, can affect the understanding of the players that may in turn affect their performance in the lab. It is not enough for the Lead Team members to just be present as observers and data collectors during the runs. They must also be prepared to answer players' questions. Each Lead Team member must have a complete understanding of the simulation interface, the scenario, and the architectures to competently answer these questions.

The scenario typically changes from experiment to experiment. Any changes to the present experiment from the previous one must be input to the simulation. This involves modifying the XS files if using DDD, or setting up batch files for MTWS. For Experiment 3, the assistance of Dr. Kleinman (UConn/NPS) and Philip Young (APTIMA) was key in ensuring that all files were correctly modified. Also, subtle changes to predetermined movement of enemy assets are required to create "alternate" scenarios to ensure that the players do not see the

same enemy actions at the same time in the scenario. Once the scenario has been translated to the simulation, the Lead Team "play-tests" each architecture and scenario to uncover any "bugs". Task balancing is necessary to make sure that no architecture is significantly more difficult to execute in the time allotted than another.

3. Player Preparation

The players used in experiments at NPS are usually involved as part of a classroom project that only comprises a fraction of the overall course objectives and time. A significant "ramping up" prior to the experiment to enable the players to perform satisfactorily in the lab is required. It falls on the Lead Team to ensure that the players are properly trained in the scenario, initial architecture, and "buttonology", which is a term for the operation of the simulation interface. Much of the information required for the players to gain a grasp on the background of the experiment can be provided in the form of handouts describing the mission (Operations Order), and the DDD tutorial, which describes how tasks are performed in the simulation. Both of these documents were used in Experiment 3 and are included in this thesis as appendices. Since these documents provide more information than is actually needed to perform in the lab, a face-to-face, focused mission briefing should be prepared by the Lead Team and delivered prior to the buttonology training. The purpose of the mission briefing is two-fold. First, it will focus the players on the mission tasks and how to perform them, and second, it will allow the players to ask questions concerning the handouts (DDD tutorial and Operations Order) prior to the buttonology training.

4. Laboratory Preparation

The unclassified DDD simulation resides in the STL at NPS and can accommodate up to seven workstations (expandable). One of the primary Lead Team functions during the experiment runs is ensuring that the correct scenario and architecture are loaded on the DDD. It is also the responsibility of the Lead Team to make sure that the player's headsets are available before the first run and that each is tested for functionality. If the players are using multiple communications nets to simulate, for example, a ground net and an air net, set up in the correct configuration for the architecture that they are playing is done by the Lead Team. For data collection, and as a backup, each run is both video- and audio-taped. For the videotape, all runs for each team should be on one tape. If possible, the counter settings should be noted on the tape's label to aid the research team when they review the tapes during their analysis. It is suggested that the camera be pointed at the observer workstation monitor if present. This will give the analysis team a picture of what is happening as they are listening to the audio. Following the experiment run, the after-action session is also videotaped.

B. CONDUCT OF EXPERIMENT

1. Aids to the Observers

Following each run, the DDD automatically saves files pertaining to the run. As a backup, to avoid data loss due to say a hard-disk failure, each run should also be saved to a 3.5" floppy disk. With all of the hardware setup tasks and the requirement for Lead Team personnel to observe and monitor each run,

it is highly suggested that the tasks required for each run be made available in checklist format. The pace of events is occasionally hectic, and a checklist would help obviate the possibility of recording instruments being overlooked or data being taped over by observers during the next run.

2. Equipment Storage

Due to security requirements, video cameras may not be left set up in the STL. Laboratory technical personnel can make locker storage space available for the overnight storage of the camera, player handouts and the 3.5" backup disks. The video and audiotapes should be handed over to the A2C2 researchers for storage.

3. Observer Workstation

As mentioned above, an observer workstation should be made available in addition to the players' workstations. The purpose of this is to allow the video camera to view the run as it collects the audio from the players. This extra station will allow the researchers, Lead Team observers, or visiting VIPs to view the experiment without disturbing the participants. This extra workstation must be "built" into the DDD software as an extra node possessing no assets. Of course, extra headsets must be made available for this station. It might be helpful to project the observer workstation image onto the large wall screen.

4. Information Distribution

The various class schedules of the Lead Team and players and inevitable experiment changes (e.g., architectures, procedures, new features of the

simulation that differ from previous years, etc.) can cause a situation where critical information is slow to filter to all experiment participants. It may be helpful for the Lead Team to use an experiment Web site to post information such as handouts, buttonology procedures, or frequently asked questions (FAQ). If this proves more trouble than it is worth, a bulletin board outside the lab could be useful for schedule, handouts, etc. Finally, it is suggested that the Lead Team leader maintain an email chain to pass information to the student class leaders.

C. FUTURE EXPERIMENTATION

1. Continuation of Experiment 3

It was determined during the analysis of data collected for Experiment 3 that a smaller more focused experiment should be run before the A2C2 research effort conducts the next major experiment. The reason for this is that there were results from the data that suggested a significant amount of learning taking place throughout the experiment (see Chapters IV and V). The training of the players was probably not equally balanced for inter-nodal coordination of assets and intra-nodal coordination. Players were relatively more adept at performing attacks autonomously, but were not as good at coordinating with another node's assets. Also, the differences due to the characteristics of the architectures and the differences due to the number of nodes in each architecture confounded the analysis effort. The specific research questions for Experiment 4 will be determined after the completion of the analysis of Experiment 3.

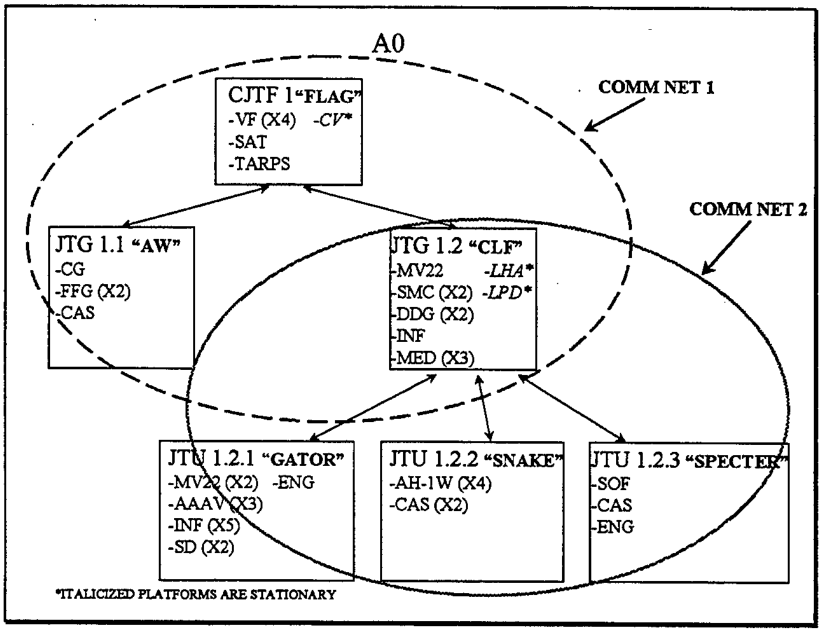
D. CONCLUSION

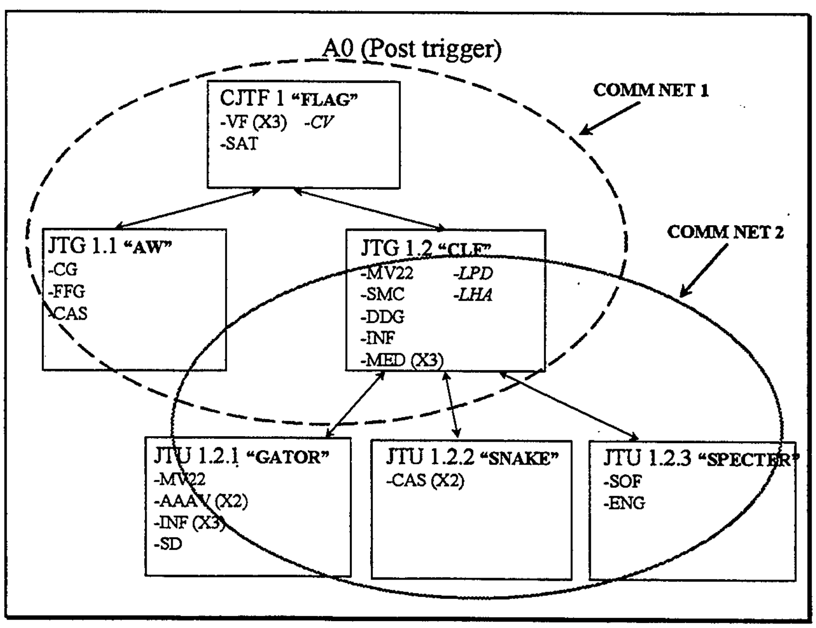
Given the difficulties with human-in-the-loop experimentation such as time constraints, learning, and simulation complexity, other strategies may allow a more efficient method of collecting valuable data when experiment objectives do not target human interactions. The complexity of the DDD simulation necessitates a large amount of time being devoted to team and individual training to allow a baseline understanding of the interface. Unfortunately, determining how much training time is sufficient is difficult to predetermine, and so far can only be determined by observing player performance during the runs. It may be possible for future experiments to draw from a pool of highly trained operators that can enact the decisions made by the players, thus eliminating the need for the player to perform as both "commander" and "private". This will more closely resemble real-world decision-making, and allow the players to concentrate on what tasks need to be performed and not how to perform them.

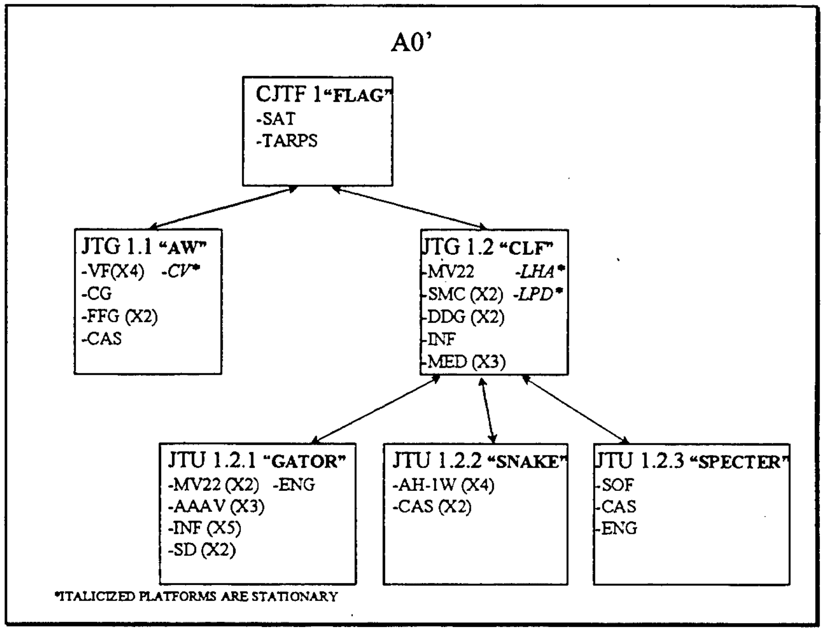
If human decision-making is not a targeted parameter of the experiment, migrating the A2C2 research project to the MAGTF Tactical Warfare Simulation (MTWS) may enable the rapid collection of architecture performance data. This simulation supports closed-loop experimentation and will allow a large sample size to be collected in a relatively short amount of time.

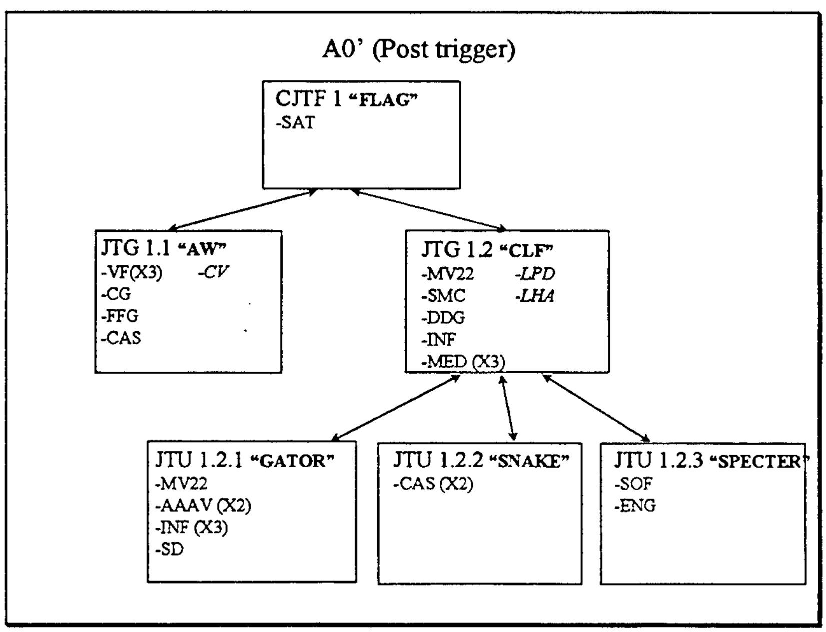
APPENDIX A. ARCHITECTURES

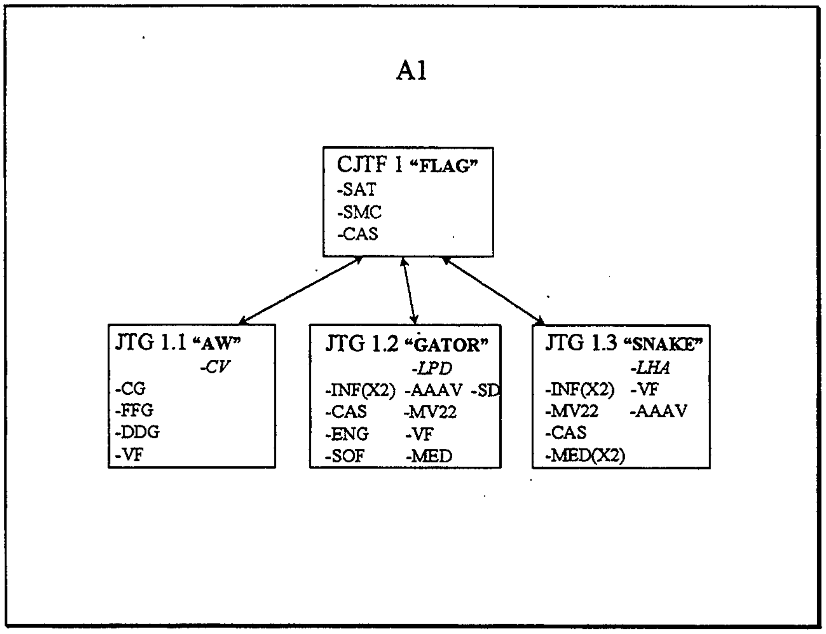
Appendix A contains the architectures that were used for Experiment 3. Teams operated under the A0 or A0' architectures for the first run. Post trigger, the teams played their "choice" and either A1 or A2. If their choice was the initial architecture, then the team played the A0 or A0' post-trigger architecture with the reduced assets.

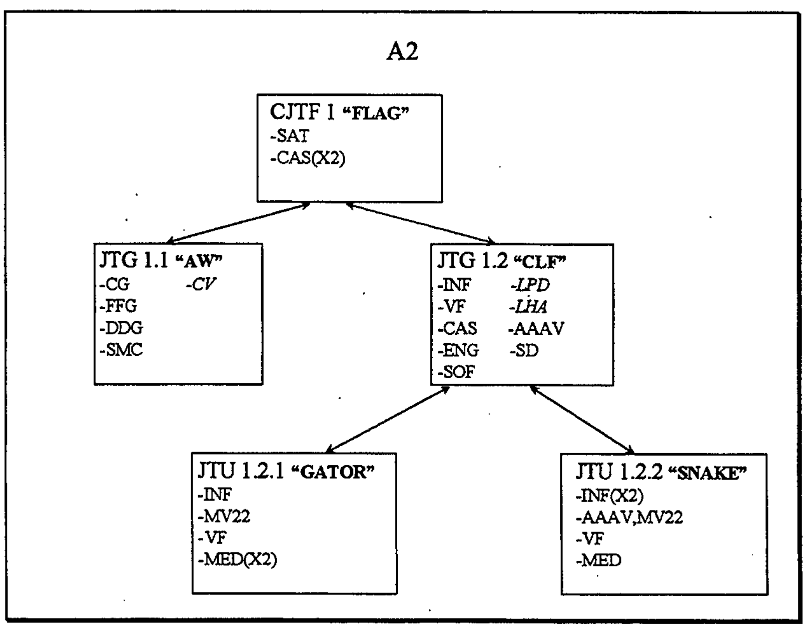





A0'
CJTF 1 "FLAG"
-SAT
-TARPS
JTG 1.1 "AW"
-VF(X4) -CV*
-CG
-FFG (X2)
-CAS
JTG 1.2 "CLF"
-MV22 -LHA*
-SMC (X2) -LPD*
-DDG (X2)
-INF
-MED (X3)
JTU 1.2.1 "GATOR"
-MV22 (X2) -ENG
-AAAV (X3)
-INF (X5)
-SD (X2)
JTU 1.2.2 "SNAKE"
-AH-1W (X4)
-CAS (X2)
JTU 1.2.3 "SPECTER"
-SOF
-CAS
-ENG
*ITALICIZED PLATFORMS ARE STATIONARY


A0' (Post trigger)
CJTF 1 "FLAG"
-SAT
JTG 1.1 "AW"
-VF(X3) -CV
-CG
-FFG
-CAS
JTG 1.2 "CLF"
-MV22 -LPD
-SMC -LHA
-DDG
-INF
-MED (X3)
JTU 1.2.1 "GATOR"
-MV22
-AAAV (X2)
-INF (X3)
-SD
JTU 1.2.2 "SNAKE"
-CAS (X2)
JTU 1.2.3 "SPECTER"
-SOF
-ENG


A1
CJTF 1 "FLAG"
-SAT
-SMC
-CAS
JTG 1.1 "AW"
-CV
-CG
-FFG
-DDG
-VF
JTG 1.2 "GATOR"
-LPD
-INF(X2) -AAAV -SD
-CAS -MV22
-ENG -VF
-SOF -MED
JTG 1.3 "SNAKE"
-LHA
-INF(X2) -VF
-MV22 -AAAV
-CAS
-MED(X2)


A2
CJTF 1 "FLAG"
-SAT
-CAS(X2)
JTG 1.1 "AW"
-CG -CV
-FFG
-DDG
-SMC
JTG 1.2 "CLF"
-INF -LPD
-VF -LHA
-CAS -AAAV
-ENG -SD
-SOF
JTU 1.2.1 "GATOR"
-INF
-MV22
-VF
-MED(X2)
JTU 1.2.2 "SNAKE"
-INF(X2)
-AAAV,MV22
-VF
-MED

APPENDIX B. DDD TUTORIAL

Appendix B is the handout given to the players explaining much of the functionality of the DDD simulation. Given to the players before formal "buttonology" training in the lab, it was intended to assist the players in becoming familiar with how to find and move the assets under their control, attack enemy units with one or more assets, or coordinate an attack with another DM's assets.

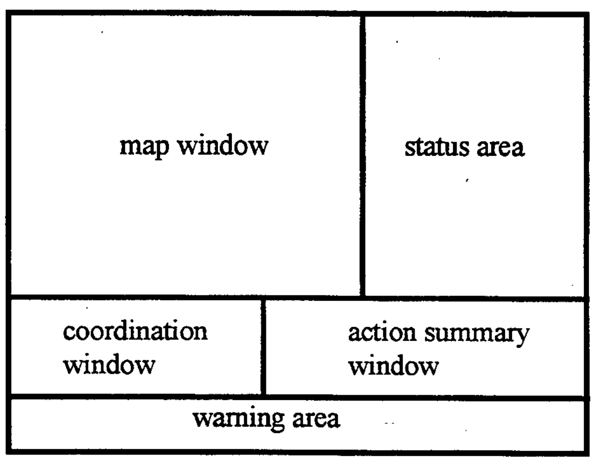
DDD Tutorial '97

This handout is organized as follows:

1. DDD Screen Layout
2. Map Icons
3. Friendly Force Icon Actions
4. Enemy Force Icon Actions

1. DDD Screen Layout

The screen is partitioned into 5 work areas: map window; status area; coordination window; action summary window; and a warning message area.



a. Status Area.

(1) **Color** of the platforms your station controls is shown by the color of the stick man figure. Also listed is the name of the station you are playing (i.e. CJTF 1, CJTG 1.1, etc.)

(2) **Time Bar**. When a friendly platform or sub-platform is selected to perform an action (i.e. launch aircraft, attack), a white arrow will appear next to

this bar showing the amount of time to complete this mission. The platform cannot perform any other action until this action is completed. In addition, above the time bar are several other pieces of information.

(3) **Mission Counter and Strength Counter**. Displays feedback on how well the entire team is doing on the scenario. The mission counter starts at zero and increments as missions are accomplished. The strength counter starts at 100 and decrements as your force strength is diminished.

(4) **Start/Refresh button.** The Start button is used only at the beginning of a scenario to start all of the stations playing. Once the scenario has begun, the button changes to Refresh. Left click on the Refresh button redraws the map and eliminates any undesired traces which may appear.

(5) **Zoom In button.** Allows the user to zoom in for a more detailed look at a particular section of the map. To zoom in, left click on the "Zoom In" button. Move the cursor over to map to where the area of interest lies. Click and hold the left mouse button and drag the cursor over the area to be zoomed. The cursor will be at the bottom left corner position of the box that will appear.

(6) **Zoom Out button.** Left click on this button returns the map to the previous map size.

(7) **Cancel button.** Left Click on the Cancel button allows the user to suspend an operation on an asset such as a move or an attack prior to completing the mission.

b. Map window.
Within the map, land is represented by squares which have a brown tint, and sea by squares which are white. Various icons appear on the map and represent friendly or enemy forces, as well as some terrain features. These icons can be manipulated with the three mouse buttons.

c. Coordination window.
Displays messages between the various players which may require some action to be taken by your station.

d. Action Summary window.
Summaries of messages or actions performed by your station will appear in this window along with some messages about the status of other friendly platforms.

e. Warning Area.
Displays warning and error messages. A beep will occur along with a warning or error message if any action performed by your station is not allowed (i.e. Attempting to attack the enemy when your unit is out of range).

2. Map Icons

a. Terrain and tasks

The following shows representations of the icons which represents terrain or tasks in the scenarios.

 **Airfield:** This airfield has attributes associated with it which must be compared to the attacking force attributes to determine if the necessary force is available. Requires a coordinated attack. The airfield takes priority over the port if they cannot be attacked simultaneously.

Port: The port , like the airfield, also has attributes which must first be determined and compared to attacking forces attributes to determine if enough combat power can be brought to bear to achieve this objective.

Hill: The hill is the commanding terrain between the port and airfield. It is surrounded by swamps on both sides which means the only way of accomplishing this mission is by using MV-22 sub-platforms, in a coordinated attack with other assets..

Bridge: The bridges are located along roads leading to the west. SOF forces must detect traffic along these roads. ID of this traffic must be done by SAT or TARPS. Once identified as enemy the lead vehicle must be destroyed along with the corresponding bridge using a combination of assets.

T **Task:** The task icon has attributes which must first be identified and then a determination made as to the best asset available to complete this task. Tasks are normally used to represent enemy objects in a given location which must be eliminated (beaches, etc.).

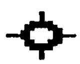
+ **Medevac:** The medevac icon is a mission which may appear after friendly ground platforms or sub-platforms engage enemy platforms . The task has attributes which must be determined. The mission is completed by attacking it with the medivac helicopter (MED sub-platform). Medevac helos have a limited on station time of 10 minutes.

Hold: The hold icon may appear after completion of a mission (i.e. attacking the hill). If this occurs the asset used to perform the mission must perform the "hold". This is done by executing an attack on the icon.

G-SOF SOF Assets: SOF have All-Terrain style vehicles which are capable of off-road movement—the only ground assets (besides the ENG platoon) which have this capability.

b. Enemy Forces.
The following section shows the icons which represent enemy forces that may or may not appear in a scenario. The text which follows each icon describes the enemy platforms capabilities and the friendly weapon of choice to use against it.

 **Artillery:** Enemy artillery pieces may pop up at various times. When they appear, they take approximately 5 minutes to set up before they are able to fire. The pieces are stored in reinforced concrete bunkers with the ammunition stored deep underground. The methods by which enemy artillery may be suppressed is through the use of Naval Surface Fire Support (NSFS), Close Air Support (CAS), or Cobra attack helicopter. NSFS can be accomplished by either the DDG or CG. Once the artillery pieces begin to move toward you, which simulates firing, you will be unable to attack them.

 **Mines:** The enemy possesses the possibility of deploying both land and sea mines. If encountered and moved through by friendly forces the friendly forces strength total will be reduced. Sea based mines may only be cleared by the use of mine clearing helicopter (SCM sub-platform). Land mines may only be cleared through the use of the engineering platoon (ENG sub-platform). Once INF detect mines, a warning will be displayed. If the INF again moves toward mines before they have been cleared, the mines will detonate, reducing overall strength.

 **Frog Missile sites:** These sites are capable of launching short range missiles containing chemical munitions. The launchers take approximately 10 minutes to set up. Suppression must be done through the use of CAS aircraft carrying precision guided munitions located on the aircraft carrier, NSFS, or Cobra attack helicopter.

 **Silkworm Missile Site:** The enemy has placed silkworm missile sites in residential areas near the port. The appearance of a silkworm site requires visual confirmation through use of a Recon (Tarps on SAT sub-platform) prior to attacking the site. The site may only be destroyed by using CAS carrying precision guided munitions. Requires coordinated attack with SAT or TARPS for laser designation.

 **SAM Site:** The enemy has placed Surface-to-Air Missile sites (as well as decoys) around the port and airfield. These sites must be identified and destroyed before air support or helo-borne forces can safely be used for the attack on the port. Must be hit with guided munitions to avoid collateral damage.

As with Silkworm sites, requires a laser designation from the SAT or TARPS. Once any air asset detects a SAM site, a warning will be displayed and the unit will stop. If the air asset again moves toward the SAM site before it has been cleared, the mines SAM will detonate in simulation of an attack on the air asset, reducing overall strength.

Submarine: The enemy submarines are Alpha class nuclear powered submarines. They can only be destroyed using the FFG platform.

Ship: The only ships the enemy possesses are fast patrol boats (PBs). These can be destroyed by using either the CG, DDG, or CAS aircraft. PBs require ID by SAT or TARPS or by very close inspection by friendly ships before they can be attacked.

Helicopter: The enemy possesses Hind helicopters capable of carrying Exocet anti-ship missiles. The friendly asset capable of destroying them are the CG, DDG, Stinger detachment (SD sub-platform), and fighters (VF sub-platform).

Aircraft: Enemy aircraft may launch attacks against the ships. Aircraft may be destroyed by using either the CG, DDG, Stinger Detachment, or fighter aircraft (VF) located on the carrier.

Tanks: Enemy tanks may be encountered along the road during the assaults on both the airfield and the port. The tanks can only be seen when within the detection range of friendly ground forces. If friendly forces move out of range the tank icon will disappear. Tanks can only be destroyed by using the Cobra attack helicopters, CAS aircraft, or a combination of CAS and another asset.

Unknown Enemy Platform: When ?? appears with an icon it must first be identified to determine what it is. The icon will have a letter designation followed by a "?". "A" denotes unknown air; "G" denotes unknown ground; and "S" denotes unknown sea. The platform must be identified with a suitable friendly platform or sub-platform. Identification of unknown ground platforms may only be accomplished using the Recon aircraft (Tarps sub-platform) from the carrier or the SAT.

c. Friendly Forces.

Friendly Platform Icon: This icon is used to represent friendly platforms in a scenario. The first letter demotes type of medium in which the platform operates. The letter "G" denotes a ground asset; the letter "S' denotes a sea

asset; and the letter "A" denotes an air asset. An additional letter and number designator will be shown on the map above the icon for further identification (i.e. CVN-01). Platform icons are color coded to show ownership.

 **Friendly Sub-platform Icon:** When launched from its parent platform a sub-platform will appear as a circle with a letter and number combination above it for further identification (i.e. INF-01). The middle of the circle will contain a letter to show the type of medium in which the platform operates. The letter "G" denotes a ground asset; the letter "S' denotes a sea asset; and the letter "A" denotes an air asset. Sub-platform icons are also color coded to show player ownership.

 **Friendly Platform/Sub-platform Busy Icon:** When a platform or sub-platform is directed to perform some mission such as attacking; launch a sub-platform; or when a sub-platform is directed to return, the icon will change to a box with a "x" in it. The platform or sub-platform cannot be directed to perform any other function until this mission is completed. At the end of the mission it will change back to its previous form. The Time Bar in the status area shows how long the asset will be unavailable.

3. Friendly Force Icon Actions.

Platforms are the major friendly forces in the scenarios such as carriers, amphibious ships, etc. Sub-platforms are smaller forces such as aircraft, Stinger detachments, engineering platoons, helicopters, etc. which are carried by a platform. The ownership of any sub-platform may or may not be the same as the owner of the platform it is being carried on.

a. To select a force, use left mouse click. The icon will be highlighted.

b. To display attributes of a platform, use middle mouse click or right mouse click and select "info on asset" from the pull down menu.
A pop-up window will list the attributes, ownership, and the number of all sub-platforms (if any) located on the platform. This window is also used to launch or request a launch of a sub-platform.

c. To launch a sub-platform that is on a platform owned by you.
In this case you can launch any sub-platform on your platform whether you own the sub-platform or not.

(1) Middle mouse click on platform.
(2) Left mouse click on the right arrow key in the line for the number of sub-platform(s) needed.
(3) Left mouse click OK.
(4) Repeat for each class of asset to be launched.

d. To launch a sub-platform owned by you, but located on a platform <u>not</u> owned by you.

(1) Middle mouse click on the platform where your sub-platform is located.

(2) Left mouse click on the arrow located in the line of the sub-platform needed until the desired number of sub-platforms to be launched is set.

(3) Left mouse click on OK.

A message will then be sent to the owner of the platform where your sub-platform is located requesting that it be launched. It is the responsibility of the platform owner to launch your sub-platform as requested. This "electronic request" is a software requirement. Verbal requests should also be used to alert the player of each request.

e. To display sensor ranges.

(1) Left mouse click on platform.

(2) Pop-up window will appear with options listed in lower portion of window. The sensor option will display four range rings around the platform:

(A) A yellow ring closest to the platform represents range of vulnerability.
(B) First black ring around platform indicates the visual identification range.
(C) Second black ring from the platform is range at which measurements can be made against the enemy (i.e. what do I need to take it out?).
(D) Outer most black ring from platform is the detection range.

(3) To turn the range rings off, middle click on the platform and left click on none.

f. To display weapon ranges.

(1) Left mouse click on platform.

(2) Pop-up window will appear with options listed in lower portion of the window. The weapons option displays a single red ring around the platform which shows the effective range of the weapon.

(3) To turn the range rings off, middle click on the platform and left click on none.

(4) IMPORTANT: Different targets have different ranges at which they can be detected and/or attacked. Match the asset type with the appropriate (air, sea, ground) rings.

g. To return a sub-platform to its platform.

(1) Right mouse click on platform.

(2) A menu will appear; select "return". This option may only be used for sub-platforms. Selecting this option will cause the sub-platform to return to the platform it originated from. The sub-platform will not move towards its originating platform, but instead will change to a box with a "x" in which simulates returning to the originating platform.

h. To move a platform.

(1) Right mouse click on platform.

(2) A menu will appear. Selecting move will cause a cross-hair type symbol to appear. Position this cross hair to the place the platform is desired to be moved and single click with the left mouse button. The platform will then move to this position. When it arrives there, it will stop until another command to move is given. Moving assets show a line extending from the asset. This line corresponds to speed and direction of movement.

I. To pursue an enemy platform.

(1) Right mouse click on platform.

(2) A menu will appear. Selecting "pursue" will cause the cursor to change to a finger. Place the finger on the enemy platform desired to be pursued, left click, and your platform will then move to intercept it until further directed.

J. To attack an enemy platform.

(1) Right mouse click on platform.

(2) A menu will appear. Select Attack. When this option is selected a question mark will appear. Place the question mark on the enemy platform to be attacked and left click. If you are in range to perform this action, a window will appear which shows the attributes of the friendly platform selected to perform the attack and the attributes of the enemy platform to be attacked. Click OK to execute the attack.

•**Coordinated Attack**: If the platform selected to attack the enemy does not have enough combat power to accomplish the mission, a

coordinated attack may be performed. It should be noted that the following explanations of how to do a coordinated attack will work only if all participating platforms are within attack range.

• **Coordinated Attack using Two Platforms**: A coordinated attack using two platforms is accomplished by first selecting one of the two platforms to perform the coordinated attack with the right mouse button, select "attack" to get the attack cursor (?) and then right clicking on the second platform performing the attack. The menu will then pop up and select the attack option again. Place the attack cursor on the platform which is to be attacked and left click with the mouse.

• **Coordinated Attack using Three or More Platforms**: To perform a coordinated attack with three or more platforms, left click on the first platform performing the attack. Then, while holding the shift key down on the keyboard, left click on all but one of the remaining platforms performing the attack. Release the shift key and right click on the final platform. The menu will pop up and select attack. The cursor will change to a question mark. Place it on the platform to be attacked and left click.

A simultaneous attack by two or more players may be needed to bring sufficient combat power to bear. These should be coordinated using the voice net. Procedures for multi-player attacks are the same as attacks shown above. However, once the window showing attributes being brought to bear/attributes required is displayed, a verbal countdown should be performed. All players contributing to the attack should click OK simultaneously.

4. Enemy Platform Actions.

a. To get known information about an enemy platform, use middle mouse click. A window appears which provides known information about the attributes of the platform (type, neutral/enemy status, attributes, etc.).

b. To identify an unknown enemy platform. (The letter in the icon is followed by a question mark.)

(1) Right mouse click on enemy platform.

(2) A menu will appear. Selecting the identify option will cause a window to pop up which shows the known attributes of the platform as seen by each player in the scenario. If a friendly platform having sensors capable of identifying the enemy platform is within sensor range the platform will be identified automatically. If not, the question mark will remain. This will be apparent by looking at the lower left hand column where the identity will be shaded from a list of possible identities. Click the fused button near the top left hand corner and then OK. The identity of the platform will then appear correctly on the map and its

icon will change to its correct identity.

The following tables give descriptions of the two letter symbols which will be the options shown when identifying an platform:

Unknown Air	Description
?	Unknown
AS	Enemy attack against ships
AG	Enemy attack against ground forces
HH	Enemy helo attack against ships
NU	Neutral
SWA	Silkworm missile in flight

Unknown Sea	Description
?	Unknown
MS	Sea mines
PB	Enemy patrol boats
SS	Enemy submarines
ML	Enemy anti-ship cruise missiles
NU	Neutral

Unknown Ground	Description
?	Unknown
HL	Ground mission of taking a hill
AP	Airport ground mission
SP	Seaport ground mission
HD	Holding or occupying ground
TK	Taking a ground mission
AT	Enemy artillery
FG	Enemy FROG launcher
SWG	Enemy Silkworm missile launche
TN	Enemy tanks, troops, or vehicles
NU	neutral
MN	Landmines

c. To request information about enemy platform via another player.

(1) Right mouse click.

(2) A menu will appear. Selecting this option will cause a menu to pop up which allows you to select an other player, or all other players from whom you wish to obtain information on the enemy platform. Select the person(s) and click OK. A message will then be sent to the person(s) notifying them that this information is requested.

d. To transfer enemy platform information to other players.

(1) Right mouse click on the enemy platform.

(2) A menu will appear. Selecting transfer option will cause a menu to pop up which allows you to select a particular individual, or all the players you wish information on the enemy platform to be sent to . Select the person(s) and click OK A message will then be sent to the person(s) selected.

e. To coordinate action against an enemy platform (other than verbal).

(1) Right mouse click.

(2) A menu will appear. The use of this option allows messages to be sent between players concerning action requests, support, or intent against an enemy platform. When selected, a menu pops up displaying options for choosing who the message is to sent to and a list of messages which may be sent. The following messages may be sent:

1. I plan to handle.
2. I plan to support.
3. I cannot handle.
4. I cannot support.
5. Can you handle ?
6. Can you support ?

Select the person the message is to be sent to, a message is to be sent, and click OK. The message will then be sent to the person selected.

f. To assign an enemy force to a friendly force player. This option may only be used if you are playing a position where you are superior to someone in the chain of command and may only be directed at those people who are subordinate to you. This option will cause a question mark to appear. Place it on the enemy platform desired to be assigned and left click. A menu will then appear which allows selecting whom in the chain of command it is to be assigned to. Left click on the person desired to assign the mission to and click OK. A message will then be sent to that person notifying them they are responsible for taking care of the mission.

APPENDIX C. PRE- AND POST-TRIGGER OPERATIONS ORDERS (OPORDERS)

Appendix C consists of the two operations orders given to the teams as part of Experiment 3. The initial oporder was given to the players before the experiment runs and explains the geopolitical situation, mission, execution instructions, and other information. The trigger oporder was presented to the teams during the planning session following the first run. This order details the loss of assets and any changes to the initial oporder. Sections in this order that do not convey changes are labeled "no change". These orders were presented in message format to add as much realism as possible.

Initial Op Order

IMMEDIATE

FROM: USCINCMED NAPLES IT
JTF 1000

TO:
CJCS WASHINGTON DC
USCINCCENT MACDILL AFB FL
USCINCLANT NORFOLK VA
USCINCEUR VAIHINGEN GE
CINCFOR FT MCPHERSON GA
USCINCPAC HONOLULU HI
USCINCTRANS SCOTT AFB IL
USCINCSTRAT OFFUTT AFB NE
COMMARFORPAC HONOLULU HI
CINCPACFLT HONOLULU HI

INFO: WHITEHOUSE SITUATION ROOM WASHINGTON DC
SECSTATE WASHINGTON DC
SECDEF WASHINGTON DC
CSA WASHINGTON DC
CMC WASHINGTON DC
CNO WASHINGTON DC

DISTR: CINC/DCINC/CCJ1/CCJ2/CCJ3/CCJ4/CCJ5/CCJ6

FOR OFFICIAL USE ONLY
OPER/REDNOSE//
MSGID/ORDER/USCINCCENT//
AMPN/SPECIAL HANDLING INSTRUCTIONS
REF/A/ORDER/CJCS/011742Z NOV 97//
REF/B/ORDER/CJCS/041142Z NOV 97//
NARR/JT STRAT CAP PLN (FY 97), CJCS ALERT ORDER//
ORDTYP/OPORD/USCINCCENT 12-97//
MAP/1015/TUNSIA//
MAP/1020/ALGERIA//
NARR/SCALE 1:100,000//
TIMEZONE/Z//

HEADING/TASK ORGANIZATION//

5UNIT

/UNITDES	/UNITLOC	/CMNTS
/USCINCLANT	/NORFOLK VA	
/USCINCEUR	/VAIHINGEN GE	
/CINCFOR	/FT MCPHERSON GA	
/USCINCPAC	/HONOLULU HI	
/USCINCTRANS	/SCOTT AFB IL	/2 TAC ARLFT SQ
		/6KC-10
/USCINCSTRAT	/OFFUTT AFB NE	/2 RC-135
/COMMARFORPAC	/HONOLULU HI	/1 MEB
/CINCPACFLT	/HONOLULU HI	
/HQ USMEDCOM FWD		/(JTF 1000)
/HQ USMEDAF (MINUS)		
/2 E-3A (AWACS)		
/HQ USNAVMED (MINUS)		
/SUPPORTING FORCES		
/COMSUPNAVFOR		
/CTG 60.1 (CVBG)		
/ARG 55.2		
/ 1 MEB		
/MPS//		

GENTEXT/SITUATION

1. (FOUO) Country Orange has attacked the friendly nation of Country Green, an U.S. ally. Attacking forces have succeeded in seizing Country Green port of Eastport. Country Green government has requested U.S. assistance in taking back port of Eastport and driving Country Orange forces from Country Green.

   A. (FOUO) ENEMY FORCES

   (1) (FOUO) See current SITREP and DIN. The port of Eastport is protected by obstructions, mines, obstacles, and the presence of hidden enemy among the port facility buildings. Two beaches approx. 5 miles south of the port may be suitable for amphibious assault. Northernmost beach (designated "North Beach") has road leading to the port. Southernmost beach (designated "South Beach") has a road leading to the airfield. Waterborne approaches to the beaches are possibly mined. Commanding terrain to north of Red Beach believed occupied by enemy Heavy Mortar Platoon with a platoon of Infantry for security. This terrain dominates both North Beach and the port. Seizure and retention of this dominant terrain should be considered essential for successful attack on Red Beach and port.

   (2) (FOUO) There are two Orange bases inland. Intel reports indicate that mobile missile forces occupy one of the bases, but it is not known which. Each base is connected by road to the port-Red Beach road and the road to each base has a bridge. Missile units from either bas will have to travel down the road and cross the bridge to bring U.S. forces to within effective range. Orange tactics and the current situation dictate that Orange send an advanced force to secure the bridge before sending any Transporter Erector Launchers (TELs) across. To prevent this, a Special Operations Force (SOF) has been inserted at an observation post (OP) near the bridges. Their mission is to determine which base the missile force occupies and

blow up the bridge leading to that base. There is a significant amount of neutral commercial traffic on the connecting roads, and while the SOF sensors can detect traffic on the far side of the bridges, they cannot discriminate between neutral commercial traffic and a hostile advance force. Air (TARPS) or space based (SAT) sensors will have to be used to establish positive hostile identification (PHID).

(3) (FOUO) Believed to be at the port, but hidden from view, is company-sized armored counterattack force that could move toward North Beach in response to any amphibious assault. Similar counterattack force may exist at airfield, which is located about 5 miles inland from the South Beach. These counterattack forces could inflict serious damage if not interdicted before they reach either beach. Off-road terrain between beach, port, airfield, and commanding terrain is swampy and treacherous; and is unsuitable for travel. Thus, all ground travel, with the exception of the SOF who are equipped with advanced All-Terrain Vehicles (ATVs), must be on the roads. It is believed that one or both of the roads, which connect the port and airfield to the beaches, will be mined. Locations of any minefields are currently unknown. Port, airfield, both roads, both beaches, and commanding terrain are located within range of artillery strong points, which must be suppressed. Northernmost strong-point can fire on North Beach and port. Southernmost strongpoint can fire on both South Beach and airfield. Artillery pieces at both strong points are housed in reinforced concrete bunkers, with ammunition stored in deep underground bunkers. It is unlikely that even concentrated air attacks will completely disable the artillery strong points. Enemy can be expected to wheel out artillery pieces (from 8 to 24 at a time), set up, sight in, and fire in under 2.5 minutes. If friendly forces can get effective NSFS on target in less than 2.5 minutes, the enemy will probably wheel their artillery pieces back into bunkers and wait until another time.

(4) (FOUO) Enemy Surface-to-Air Missile (SAM) sites and decoys have been erected around the port and airfield. The SAM sites must be identified and destroyed before air support or helo-borne forces can safely be used for the attack on the port. To minimize collateral damage, the sites must be hit with guided munitions.

(5) (FOUO) Enemy also has several Frog Missile Launchers (SCUD-like) capable of carrying chemical munitions. Frogs are believed to be hidden in the vicinity of both artillery strongpoints. They can emerge from covered positions, prepare warheads, and fire missiles within 4 minutes. Past experience has shown that Frog crews are more stalwart than artillery crews. They will continue to prepare and launch their missiles even if they are being suppressed by NSFS or artillery.

(6) (FOUO) Submarine threat to U.S. Naval forces is considerable. Enemy Alfa-Class submarines are known to be in the area. To protect the fleet, these submarines must be detected and destroyed.

(7) (FOUO) Enemy possesses HIND Helicopters, and has demonstrated the capability to launch anti-ship missiles from its helicopters. The only significant capability the ARG or CVBG possesses against these helicopters is one Stinger Platoon.

(8) (FOUO) The enemy has significant air strike capability, and can launch anti-ship missiles from most of its strike aircraft.

(9) (FOUO) The enemy's Maritime Special Forces also possess numerous fast patrol boats (PBs), that can either fire very potent torpedoes, be loaded with

explosives for suicide missions, or carry troops and supplies to reinforce Orange forces. These can be engaged and destroyed by the CG, DDGs, FFGs, fighters, and Cobras. But, they have been camouflaged to resemble a type of commercial coastal craft common in the area, and they are known to travel at the same speed as coastal traffic to avoid giving away their identity. These PBs must be identified by either SAT, TARPS, or very close inspection by friendly surface platforms before they can be engaged. There are two popular coastal trade routes between the mainland and a large island to the east. One route goes to the north of Green and passes close to a small inlet which could support offloading of troops and supplies to Orange forces occupying the port area. The other route passes south along the east coast and passes close to a beach south of the airfield, which could support offloading of troops and supplies to reinforce Orange forces around the airfield. Maritime traffic along these routes, and in the region overall, must be positively identified to ensure the destruction of all hostile boats while avoiding attacking neutral shipping.

(10) (FOUO) There is also a Silkworm threat along the coastline. These Silkworm missiles were placed in residential neighborhoods by the enemy because they knew U.S. planners would be reluctant to target residential areas. Accordingly, if U.S. forces want to target a Silkworm launcher, they must first get positive confirmation of its presence, using reconnaissance assets (TARPS, SOF, Satellite). Any strike must use precision guided munitions (CAS).

B. (FOUO) FRIENDLY FORCES. JTF will be comprised primarily of assets organic to Mediterranean Command (MEDCOM). A theater-level JFACC and other friendly forces are operating against the enemy in Country Green, but not in concert with the JTF. This forces the requirement to identify contacts prior to attacking to ensure friendly and neutral forces are not destroyed.

(1) (FOUO) JTF will consist of one Aircraft Carrier Battle Group (CVBG), and a Amphibious Ready Group (ARG) with embarked Marines. The ARG will be composed of 1 LHA and 1 LPD. CVBG will be composed of 1 CV, 1 AEGIS cruiser, 2 DDGs, and 2 FFGs.

(2) (FOUO) The CVBG and ARG aircraft available to support the JTF are 4 sections of F-14s, 4 sections of F/A-18s, and 1 TARPS equipped F-14. The F/A-18s from the CV are equipped with Laser Guided Bombs (LGBs) and can attack Frog missile sites or confirmed Silkworm sites, or they can be used to provide Close Air Support (CAS) for friendly ground units. The F-14s can be used for Anti-Air Warfare (AAW) and for Combat Air Patrol (CAP). The F-14 TARPS can be used for reconnaissance missions only.

(3) (FOUO) In addition to TARPS, the JTF has access to an imagery satellite (SAT platform) which can provide continuous wide-angle "detection" coverage throughout the objective area. High-resolution "identification" coverage is available for a small (movable) area.

(4) (FOUO) Two DDGs will be in position to provide NSFS. Small Minesweeping Craft (SMCs) are attached to the ARG to clear sea mines if detected.

(5) (FOUO) The Marine amphibious forces are embarked on the ARG. The ARG is composed of three Advanced Amphibious Assault Vehicle (AAAV)-mounted rifle companies, three V-22 Osprey-mounted heliborne rifle companies, 4 sections of AH-1W SeaCobras (indivisible), two mineclearing boats (SMCs), two engineer platoons, and three of MEDEVAC helicopters. Engineers must be used to breach any minefields encountered by JTF ground forces. Cobras are the only JTF assets which

by themselves are effective against armored formations. Two Stinger Detatchments will provide a close-in anti-air capability.

(6) (FOUO) Ground forces have unmanned aerial vehicles (UAVs) flying in support for the duration of the operation. Continuous live feed will be fed to the Common Operational Picture (COP) available to all friendly forces.

GENTEXT/MISSION

2. (FOUO) On order, JTF 1 ground forces will seize and defend Country Green Port of Eastport, to allow introduction of follow on forces in support of Country Green government troops. Sea-based forces will support amphibious assault with CAS, naval gunfire, and air defense assets to defend the CVBG and ARG from air, surface, and subsurface threats.//

GENTEXT/EXECUTION/

3. (FOUO) CONCEPT OF OPERATIONS

A. GROUND. The SOF will be inserted prior to the commencement of the amphibious landings. One AAAV-mounted rifle company will land on each beach near-simultaneously. As a prerequisite to this, one heliborne rifle company will secure the commanding terrain overlooking Red Beach and the port in a coordinated attack. Once BOTH beaches and commanding terrain are secure, the two AAAV-mounted companies will proceed on foot down the roads from their respective beaches, clearing minefields with engineer platoons, killing counterattack forces with Cobras, and conducting MEDEVACS as necessary. The roads must be cleared prior to attacking the port or airfield. The SOF should conduct surveillance to locate the enemy missile force and destroy the applicable bridge, then proceed as directed to assist in assaults on the port and airfield. The UAVs will keep the artillery strong points and the suspected FROG sites under constant surveillance, so that NSFS or CAS assets can be brought to bear immediately if they are needed. Once the roads have been cleared, the AAAV-mounted companies will take the port and airfield. A heliborne company will assist the company attacking the airfield. It is important that once the AAAV-mounted companies land on the beach, the airfield and port be taken as quickly as possible, before the enemy has a chance to organize his defense and send reinforcements. It is desired that final assaults on the airfield and port occur simultaneously, in order to present the enemy commander with the most confusing environment possible. However, if one attack must occur before the other, the airfield has the priority. If the airfield attack is held up for any reason, the port attack should wait to retain the synergism of concurrent attacks. If the port attack is held up, the airfield attack should go forward.

B. MARITIME. Due to hydrographic limitations, the ARG and the CVBG will have to be significantly separated during the operation, enough to preclude them from being under a Joint Air Defense umbrella provided by the AEGIS Cruiser. Thus, the AEGIS Cruiser will remain with the CVBG, but will position itself so that it can rapidly move from the CVBG to the ARG if that becomes necessary. Additionally, the two DDGs are inshore, providing NSFS support, while the FFGs are primary ASW platforms for the CVBG. The FFGs performing ASW will, like the AEGIS Cruiser, position themselves so that they can quickly move to support the ARGs if that is necessary. The frigates, AEGIS cruiser, and destroyers can attack or be attacked by the enemy patrol boats. The ARGs will launch the Marines for the initial assault on Ted and Blue Beaches at the commencement of the operation, and will launch the minesweeping boats, SeaCobras, MEDEVAC helos, the air assault for the attack on the airfield, etc. when called to do so. The destroyers will provide NSFS to suppress enemy artillery ashore and for other missions when requested to do so. If a suspected Silkworm launcher is detected, TARPS, SOF, or Satellite must identify it before it can be destroyed. Silkworm and SAM sites require a coordinated laser

designation in order to achieve a perfect attack. A Silkworm launcher detected at the northernmost site threatens the CVBG, and one at the southermost site threatens the ARGs. SAM sites protect the port and airfield.

4. (FOUO) FIRST TASK ASSIGNMENT CLF. On order of the JTF 1, land two AAAV-mounted companie on Red Beach and Blue Beach concurrently. Simultaneously seize commanding terrain to the north of Red Beach with one heliborne company. Once the beach and commanding terrain are secure, attack along the roads from the beaches to the port and airfield with the AAAV-mounted companies, clearing minefields with the attached Engineer Platoon, killing counterattack forces with assigned Cobras and conducting MEDEVACS as necessary. Once the roads have been cleared, conduct a simultaneous coordinated attack on the port and airfield with your AAAV-mounted companies and your heliborne companies.

5. (FOUO) SECOND TASK ASSIGNMENT CVBG. Support the CLF and subordinates by launching requested assets and providing fighter support.

6. (FOUO) THIRD TASK ASSIGNMENT ARG. On order of JTF 1, ARG will initially clear mines from the beaches with the Minesweeping Boats. ARG will launch 3 Companies of Marines for the initial assault on Red and Blue Beaches and the hill. The ARG will launch the Cobras, MEDEVAC helos, the heliborne company for the attack on the airfield. ARG will also, with NSFS DDGs, suppress artillery positions.

7. (FOUO) COORDINATING INSTRUCTIONS.

A. (FOUO) This order effective for planning upon receipt and execution on order.

B. (FOUO) Dirlauth for planning and operations with Info CJCS and CINCMED.

C. (FOUO) ROE will be per CINCMED OPLAN 1234.

D. (FOUO) Friendly forces will have a UAV (launched from the ARG) airborne for the duration of the operation. The UAV's will keep the artillery strong-points and the suspected FROG sites under constant surveillance, so that NSFS or CAS assets can be brought to bear immediately if needed.

E. (FOUO) If the airfield attack is held up for any reason, the port attack will be delayed to retain the synergism of concurrent attacks. If the port attack is held up, the airfield attack will go forward.

F. (FOUO) The attack on the airfield has priority, because enemy buildup/sustainment of forces can be most quickly and effectively achieved through air transport.

GENTEXT/ADMIN AND LOG/

8. (FOUO) Per CINCMED OPLAN 1234, as amended herein.//

GENTEXT/COMMAND AND SIGNAL/

9. (FOUO) USCINCMED is supported CINC.

10. (FOUO) CJTF 1 is on-the-scene Commander and will exercise OPCON of advance forces until HQ USCINCMED FWD is activated.

11. (FOUO) Command relationships as outlined in Annex J, CINCMED OPLAN 1234.

12. (FOUO) Communications guidance as outlines in Annex K, CINCMED OPLAN 1234 as amended herein.//

AKNLDG/Y//

DECL/OADR//

Trigger Op Order

O XXXXXX NOV 97

FROM: USCINCMED NAPLES IT//

TO: JTF ONE//
All action addresses the same//

INFO: Info Address remain unchanged//

DISTR: SAME//

EXERCISE EXERCISE EXERCISE

BT

FOR OFFICIAL USE ONLY
OPER/REDNOSE//
MSGID/ORDER/USCINCMED //
AMPN/SPECIAL HANDLING INSTRUCTIONS
REF/A/DOC/USCINCMED/JUN97//
REF/B/ORDER/CJCS/041142Z NOV 97//
NARR/USCINCMED OPLAN 1234, CJCS ALERT ORDER//
ORDTYP/OPORD/USCINCMED 12-97//
MAP/1015/GREEN//
MAP/1020/ORANGE//
NARR/SCALE 1:100,000//
TIMEZONE/Z//

HEADING/TASK ORGANIZATION//

GENTEXT/SITUATION

1. (FOUO) Country Orange has attacked the friendly nation of Country Green, an ally of the U.S. Orange forces have seized Country Green port of Eastport and International Airport. Country Green government has requested U.S. assistance in taking back port of Eastport and driving Country Orange forces from Country Green.

2. Growing tensions between Armenia and Azerbazaan have erupted and fighting has been reported in and around the Turkish border. As a precaution to further escalation, many assets designated for CJTF One support have been redeployed to support the Armenia-Azerbazaan crisis (See attached assets list for changes in force structure). Commander shall review revised force structure and submit commander assessments to CJTF Four in 24 hours.

A. (FOUO) ENEMY FORCES

(1) (FOUO) No change.

(2) (FOUO) Change: Space-based (SAT) sensors will have to be used to establish positive hostile identification (PHID) on Silkworm and SAM sites.

(3) (FOUO) Change: Revised Intelligence estimates that enemy artillery can set up and fire in approximately 5 minutes. If friendly forces can get effective NSFS on target in less than 5 minutes, the enemy is expected to move their artillery pieces and back into bunkers and wait until another time.

(4) (FOUO) No change.

(5) (FOUO) Change. Revised Intelligence estimates that enemy FROG launchers can set up and fire in approximately 5 minutes.

(6) (FOUO No change.

(7) (FOUO) Change: Hinds may be engaged by fighters (VF) from the CV or the Stinger Detachment (SD) from the ARG.

(8) (FOUO) No change.

(9) (FOUO) The enemy's Maritime Special Forces possess numerous fast patrol boats, that can either fire modern torpedoes, be loaded with explosives for suicide missions, or carry troops and supplies to reinforce Orange forces. Enemy frequently camouflages PCs to resemble commercial coastal craft common in the area. PCs will travel at the same speed as coastal traffic to avoid disclosing their identity. Once identified, Patrol Boats shall be engaged and destroyed (by the CG, DDG, FFG, and CAS). There are two popular coastal trade routes between the mainland and a large island to the east. One route goes to the north of Green and passes close to a small inlet which could support offloading of troops and supplies to Orange forces occupying the port area. The other route passes south along the east coast and passes close to a beach south of the airfield, which could support offloading of troops and supplies to reinforce Orange forces around the airfield. Maritime traffic along these routes must be positively identified to ensure the destruction of all hostile boats. Neutral shipping levels remain high.

(10) (FOUO) There is a Silkworm threat along the eastern coastline. Silkworm missiles are located in residential neighborhoods to avoid U.S. targeting. To strike a Silkworm launcher requires target verification using reconnaissance assets (SOF, Satellite) and use of precision guided munitions (CAS). Intelligence estimates that Silkworm crews can setup and fire in about 9 minutes.

B. (FOUO) FRIENDLY FORCES.(Only changes reflecting revised force structure)

(1) (FOUO) Joint Task Forces: see attached force list. All CAS is now represented by F/A-18 aircraft from the CV. The F/A-18s from the CV are equipped with Laser Guided Bombs (LGBs) and can attack Frog, SAM, and Silkworm missile sites, or they can be used to provide Close Air Support (CAS) for friendly ground units. The F-14s can be used for Anti-Air Warfare (AAW) and for Combat Air Patrol (CAP). There are NO F-14 TARPS for reconnaissance missions.

(2) (FOUO) In lieu of TARPS, the JTF has access to imagery satellites that can provide continuous wide-angle "detection" coverage throughout the objective

area. Overhead directional high-resolution target identification capability is available for small (movable) areas.

(3) (FOUO) The Marine amphibious forces are embarked on the ARG. The ARG is composed of Advanced Amphibious Assault Vehicle (AAAV)-mounted rifle companies, V-22 Osprey-mounted heliborne rifle companies, minesweeping boats (SMCs), engineer platoons, and MEDEVAC helicopters. Engineers must be used to breach any minefields encountered by JTF ground forces and assist in blowing up the correct bridge. Stinger Detachments will provide a close-in anti-air capability. Note, there are no AH-1 SeaCobras available.

GENTEXT/MISSION

2. (FOUO) Change: Order to ground forces will be given by CJTF One.//

GENTEXT/EXECUTION/

3. (FOUO) CONCEPT OF OPERATIONS

A. GROUND. No Cobras available to kill counterattack forces, all else remains the same.

B. MARITIME. Due to hydrographic limitations, the ARG and the CVBG will have to be significantly separated during the operation, enough to preclude them from being under a Joint Air Defense umbrella provided by the AEGIS Cruiser. Thus, the AEGIS Cruiser will remain with the CVBG, but will position itself so that it can rapidly move from the CVBG to the ARG if that becomes necessary. Additionally, the DDG is inshore, providing NSFS support, while the FFG is primary an ASW platform for the CVBG. The FFG performing ASW will, like the AEGIS Cruiser, position itself so that it can quickly move to support the ARGs if that is necessary. The frigate, AEGIS cruiser, and destroyer can attack or be attacked by the enemy patrol boats. The ARGs will launch the Marines for the initial assault on Red and Blue Beaches at the commencement of the operation, and will launch the minesweeping boats, MEDEVAC helos, the air assault for the attack on the airfield, etc. when called to do so. The destroyer will provide NSFS to suppress artillery strong points ashore and for other missions when requested to do so. The CVBG will provide CAP aircraft. If a suspected Silkworm launcher is detected, SOF, or Satellite must positively identify it before it can be destroyed. A Silkworm launcher detected at the northernmost site threatens the CVBG, and one at the southernmost site threatens the ARG.

4. (FOUO) FIRST TASK ASSIGNMENT Landing Force. On order of the JTF, land one AAAV-mounted company each on Red Beach and Blue Beach near-simultaneously. Prior to taking the beaches, seize the commanding terrain to the north of Red Beach with one heliborne company. Once the beaches and commanding terrain are secure, attack along the roads from the beaches to the port and airfield with the infantry companies, clearing minefields with the attached Engineer Platoon and conducting MEDEVACS as necessary. Once the roads have been cleared, conduct a simultaneous coordinated attack on the port and airfield with your infantry companies and your heliborne companies.

5. (FOUO) THIRD TASK ASSIGNMENT ARG. On order of JTF, ARG will initially clear mines from the beaches with the Minesweeping Boats. ARG will launch Marines for the initial assault on Red and Blue Beaches and the hill. The ARG will launch the MEDEVAC helos when called to do so. ARG will also, with NSFS DDG, suppress artillery strong points ashore.

6. (FOUO) COORDINATING INSTRUCTIONS. A through M No Change.//

7. GENTEXT/ADMIN AND LOG/   No Change.//

8. GENTEXT/COMMAND AND SIGNAL/

(FOUO) One change: CJTF 1 is on-the-scene Commander and will exercise OPCON of advance forces until HQ USCINCMED FWD is activated.

AKNLDG/Y//

BT

APPENDIX D. PLAYER REFERENCE HANDOUTS

Appendix D consists of the reference handouts given to the players before each run. These were designed to allow a rapid transition to the unfamiliar architectures in the post trigger runs. Since many assets "owned" by DMs began the scenario on ships controlled by another DM, the players were required to request that their assets be launched before they could take control of them. Another handout showed each of the mission tasks and the required asset combinations to accomplish the task.

OWNER	ASSETS	STARTING POS.	MUST REQUEST LAUNCH FROM...
FLAG	VF (X3) SAT	CV "	N/A "
AW	CG FFG CAS	water " CV	N/A " FLAG
CLF	SMC DDG INF MED (X3)	water " LHA (MV22) LHA(2),LPD(1)	N/A " " "
GATOR	INF (X3) SD ENG	LPD(MV22,AAAV) LHA (AAAV) LPD LPD	CLF CLF CLF
SNAKE	CAS (X2)	CV	FLAG
SPECTER	SOF ENG	land LHA	N/A CLF A0post

Example of asset starting position handout

TASK	REQUIREMENTS(UNITS)	CAPABLE ASSETS
Beach/Hill*	GASLT (10) FIRES (14) ARMOR (12)	{INF + 2 AIR (CAS or AH-1)} {INF + AIR + DDG} {2INF +CAS}
Airport/Seaport	GASLT (20) FIRES (10) ARMOR (4)	{2INF + AIR} {2INF + (DDG or CG)}
SAM and Silkworm sites**	LASER DESIG (6) ARMOR (8)	{(SAT or TARPS or SOF) + CAS***}
Lead vehicle of enemy relief column	SOF MUST DETECT. ID BY SAT OR TARPS. LASER DESIG (4) ARMOR (8)	{(SAT or TARPS or SOF) + AIR}
Bridge	FIRES (8) ARMOR (6) LASER DESIG (10) MINES (4)	{SOF + CAS + ENG}
*Must use heliborne INF to take the Hill ** Must be ID'ed by SAT, SOF, or TARPS ***Must use CAS w/LGBs		

Task and required asset handout

DDD '97 ASSET DESCRIPTION

CV: Aircraft Carrier. The ship which carries the fighter assets and the CJTF.

CG: Guided Missile Cruiser. A ship equipped with the AEGIS radar system. It's primary role is Anti-Air Warfare, defending the carrier.

DDG: Guided Missile Destroyer. A shipped equipped with two 5"/54 guns. The role of the DDG in DDD is naval surface fire support for the landing forces.

FFG: Guided Missile Frigate. The role of the FFG in DDD is Anti-Submarine Warfare (ASW).

LPD: Amphibious Transport Dock. A ship whose purpose in DDD is transporting and off-loading the ground forces.

LHA: Amphibious Assault Ship. A "helicopter carrier". The purpose of this ship in DDD is to launch helicopters and AV-8B Harriers in support of the ground forces.

SOF: Special Operations Forces. Elite teams of men, inserted behind enemy lines to observe enemy movement and destroy the bridge.

TARPS: Tactical Air Reconnaissance Pod System. A system attached to VF aircraft for use in tactical intelligence gathering.

AAAV: Advanced Amphibious Assault Vehicle. A vehicle used to carry landing forces ashore. Can also be used for troop transport on land, but, for the purposes of DDD, are constrained to traveling on roads.

MV-22: Tilt-rotor troop transport aircraft, capable of forward flight like an airplane and take-offs and landings like a helicopter. Used to air-transport troops ashore.

ENG: Combat engineers. Ground forces which are, in DDD, used for the clearing of landmines.

SMC: Minesweepinig ships. Small ships used, in DDD, for clearing mines at sea.

VF: Fighter aircraft from the CV. For DDD, they have an air-superiority mission and are charged with defending the carrier battle group.

CAS: AV-8B Harrier aircraft from the ARG. These aircraft are capable of vertical and short take off and landing. In DDD, these assets provide Close Air Support for ground forces.

AH1 SeaCobra: Attack helicopter used to provide airborne fire support for ground forces. Possess an anti-armor capability.

SAT: Satellite "beam" steered by friendly forces in order to identify enemy assets.

INF: Infantry companies.

MED: Medical evacuation (medevac) helicopters.

SD: Stinger detachment. Used in DDD to counter close-in air threats to the battlegroup/ARG.

Example Task Assignments for JTF1. You may deviate from this as desired.

CJTF 1: Callsign "Flag". You are the team lead, and are tasked with coordinating subordinate missions. Keeping the "big picture" is essential for this mission. You will control the identification platforms (SAT and TARPS) essential for attacking some targets.

JTU 1.1: Callsign "AW". Conduct Anti-Air Warfare (AAW) with the CG and Anti-Submarine Warfare (ASW) with the frigates throughout the operation. CAS is provided to deal with identified enemy patrol boats and to assist the Landing Force in operations ashore.

JTU 1.2: Callsign "CLF". Support the Landing Force by clearing any mines which may impede the AAAVs. Use the heliborne infantry to seize the hill which dominates the North Beach. DDGs are available to support the Landing Force. Conduct Medevac operations as required.

JTG 1.2.1: Callsign "Gator". Once the hill is secured, land one AAAV-mounted infantry company on each beach near-simultaneously. After securing each beach, attack along both roads to the Airport and Seaport, clearing any minefields with the Engineers. Call in air support and Medevacs as necessary. Once the roads and SAMs have been cleared, conduct attacks on the Port and Airport. These attacks should be as nearly simultaneous as possible in accordance with the Oporder.

JTG 1.2.2: Callsign "Snake". Support the Landing Force with Cobras and CAS as required.

JTG 1.2.3: Callsign "Specter". Launch the SOF force from their base immediately to begin detecting vehicles along the roads to the enemy bases to the west. Coordinate with Flag to identify the Lead Vehicle of the enemy missile force reportedly heading towards the landing area. Once identified, destroy the vehicle and conduct a coordinated attack on the corresponding bridge with CAS and Engineers.

Sample tasking for each node

APPENDIX E. TASK VALUES, TASK REQUIREMENTS VECTORS, AND ASSET RESOURCE VECTORS

Appendix E shows the details of the DDD XS files for Experiment 3. Requirements vectors show how much of what resource is required to accomplish the specific task. The friendly asset resource table shows how much of each resource vector each asset possesses.

LAND AREA from (0,40) to (30,100)
ISLAND from (50,40) to (70,60)

VERSION 3.1
11-5-97

PENETRATION ZONES
zone 0 (25, 45) seaport
zone 1 (25, 60) high ground
zone 2 (28, 73) beach A (north)
zone 3 (28, 83) beach B (south)
zone 4 (05, 95) airfield
zone 5 (70, 15) fleet north (CV)
zone 6 (64, 75) fleet south (ARG)
zone 7 (15, 40) enemy resupply port1
zone 8 (30, 95) enemy resupply port2

PLATFORM CLASSES
class 0: (GOD) test platform for overall coverage
class 1: (DDG) destroyer, N=2
class 2: (FFG) Frigate, N=2
class 3: (CG) Cruiser N=1
class 4: (CV) aircraft carrier N=1
class 5: (LSD) Amphibious assault ship
class 6: (LHA) Amphibious assault ship
class 7: (LPD) Amphibious assault ship
class 8: (ENG) amphib based engineering company helicopter, N=2
class 9: (INF) ground born infantry company, N=6
class 10: (SD) stinger detachment, N=2
class 11: (AH1) amphib based attack helicopter (Cobras), N=4
class 12: (CAS) carrier based F-18 close air support, N=4 at any one time
class 13: (VF) carrier based F-14 attack a/c, N=4 at any one time
class 14: (MED) amphib based medivac helicopter, N=2
class 15: (SMC) mine-clearing ship, N=2
class 16: (TARP) recon aircraft - carrier based, N=1
class 17: (AAAV) amphib based troop carrier sea-going, N=3
class 18: (MV22) amphib based troop carrier helicopter, N=3
class 19: (SAT) satellite, N=1
class 20: (SOF) Special Operations Force, N=1
class 21: (BASE) Base from which SOFs operate

SUBPLATFORM DISTRIBUTION:
aircraft carrier carrying: fighters(DMx), CAS(DMy), recon a/c(DMz)
platform subplatform 4 3

VF	CAS	TARP
8	8	1
x	y	z

#

```
# amphib with: eng/med/ AAAV/MV2/SD, Cobras
platform subplatform  6  6
     ENG  AH1  MED MV22  AAAV  SD
     1     2     1     1     2     1
     a     b     c     d     e     f
#
# amphib with: eng/med/ AAAV/MV2/SD, Cobras
platform subplatform  7  6
     ENG  AH1  MED MV22  AAAV  SD
      1     2     1     2     1     1
      a     b     c     d     e     f
#
# AAAV carrying: ground based rifle company
 platform subplatform  17  1
      INF
      1
      -1
#
# MV22 carrying: ground based rifle company
 platform subplatform  18  1
      INF
      1
      -1
# BASE carying: SOF
 platform subplatform  21  1
      SOF
      1
      -1
#
TASKS:
# task 0: ground mission (hills)
# task 1: ground mission (airport)
# task 2: ground mission (seaport)
# task 3: ground mission (Holding/occupying)
# task 4: ground mission (Taking N. beach)
# task 5: ground contact (artillery)
# task 6: ground contact (Frog launchers)
# task 7: ground contact (silkworm anti-ship missile site)
# task 8: ground contact (mines)
# task 9: sea contact (mines)
# task 10: air contacts (air-sea attackers)
# task 11: air contacts (air-ground attackers)
# task 12: air contacts (SAM site)
# task 13: air contacts (dummy SAM site)
# task 14: ground contact (enemy tanks, vehicles, armor, etc.)
# task 15: ground contact (dummy silkworm site)
# task 16: sea contact (Patrol boats)
# task 17: sea contact (enemy submarines)
# task 18: sea contact (neutral commercial sea traffic)
# task 19: sea contact (neutrals - looking like Patrol boats)
# task 20: medevac
# task 21: air contacts (neutral COMMAIR, etc.)
# task 22: air contacts (unstoppable missile)
# task 23: Taking S. Beach
# task 24: ground contact (missile-carrying transport = lead vehicle)
# task 25: ground contact (neutral - looking like missile transport)
```

task 26: blow up (WRONG) bridge
task 27: blow up bridge

ATTRIBUTES:

1. VALUE
2. TIME
3. AIR
4. ASUW
5. ASW
6. GASLT
7. FIRES
8. ARMOR
9. ENEMY

RESOURCES: (r1-r6 = a3-a8)

1. AIR
2. ASUW
3. ASW
4. GASLT
5. FIRES
6. ARMOR
7. *HOLD*
8. *MINE*
9. *MED*
10. *IDES*

NOTE-0: All numbers are subject to minor changes

NOTE-1: 'ENEMY' attribute is used to distinguish hostile from neutrals

CURRENT ASSIGNED VALUES (ATTRIBUTES) PLUS RESOURCES REQUIRED

TASK	VALUE	TIME	AIR	ASUW	ASW	GASLT	FIRES	ARMOR	ENEMY	*HOLD*	*MINE*	*MED*	*IDES*
0 HILL	**10**	10	0	0	0	10	14	12	0	0	0	0	0
1 AIRPRT	**30**	30	0	0	0	20	10	4	0	0	0	0	0
2 SEAPRT	**30**	30	0	0	0	20	10	4	0	0	0	0	0
3 HOLD	10	90+	0	0	0	0	0	0	0	10	0	0	0
4 BEACH	**10**	10	0	0	0	10	14	12	0	0	0	0	0
5 ARTY	2	10	0	0	0	0	0	5	1	0	0	0	0
6 FROG	10	10	0	0	0	0	0	5	1	0	0	0	0
7 SILK(H)	15	15	0	0	0	0	0	8	1	0	0	0	6
8 GMINE	5	20	0	0	0	0	0	0	1	0	5	0	0
9 SMINE	10	20	0	0	0	0	0	0	1	0	10	0	0
10 AIR(Sea)	15	20	5	0	0	0	0	0	1	0	0	0	0
11 AIR(Gnd)	4	20	5	0	0	0	0	0	1	0	0	0	0
12 SAM(H)	10	20	0	0	0	0	0	8	1	0	0	0	6
13 SAM(N)	*10*	*20*	*0*	*0*	*0*	*0*	*0*	*8*	*0*	*0*	*0*	*0*	*0*
14 TANK(H)	5	10	0	0	0	0	0	10	1	0	0	0	0
15 SILK(N)	*15*	*15*	*0*	*0*	*0*	*0*	*0*	*8*	*0*	*0*	*0*	*0*	*0*
16 SEA(Pb)	15	10	0	3	0	0	0	0	1	0	0	0	0
17 SEA(Sub)	15	10	0	0	10	0	0	0	1	0	0	0	0
18 SEA(N)	10	10	0	3	0	0	0	0	0	0	0	0	0
19 SEA(PbN)	*15*	*10*	*0*	*3*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
20 MEDVC	5	60	0	0	0	0	0	0	0	0	0	10	0
21 AIR(N)	15	20	5	0	0	0	0	0	0	0	0	0	0
22 MISSLE	15	20	30	0	0	0	0	0	1	0	0	0	0
23 DUMMY	0	0	0	0	0	0	0	0	0	0	0	0	0
24 GTL(H)	15	10	0	0	0	0	0	8	1	0	0	0	4
25 GTN(N)	*15*	*10*	*0*	*0*	*0*	*0*	*0*	*8*	*0*	*0*	*0*	*0*	*0*
26 BRIDGE	15	20	0	0	0	0	8	6	0	0	4	0	10
27 BRIDGE	**15**	20	0	0	0	0	8	6	0	0	4	0	10

NOTE-1: All tasks in italics (Neutral/dummy) may be distinguished from their non-italic (threat) counterpart by measuring the "enemy" attribute, a9. For these sets of tasks, ONLY the SAT, TARP and SOF can measure a9. However, only the SOF has a reasonable range to detect the presence of tasks of class 24 & 25, but the SOF cannot measure a9 for these tasks!

NOTE-2: If any platform gets sufficiently close to any unknown task the task's identity will become known (as long as the platform's ID range is > 0). For sea contacts, the CV and ARG ships have good ID zones; CG,FFG, DDG have small ID zones (~ same as their "be-attacked" zones).

CURRENT ASSIGNED VALUES (RESOURCES)

ID	Type	Name	Vel	air	asuw	asw	ga	fire	armor	hold	mine	med	IDes
0	A	GOD	0.99	50	50	50	50	50	50	50	50	50	50
1	S	DDG	0.20	10	10	1	0	9	5	0	0	0	0
2	S	FFG	0.20	1	4	10	0	4	3	0	0	0	0
3	S	CG	0.20	10	10	1	0	9	2	0	0	0	0
4	S	CV	0.00	0	0	0	0	0	0	0	0	0	0
5	S	LSD	0.00	0	0	0	0	0	0	0	0	0	0
6	S	LHA	0.00	0	0	0	0	0	0	0	0	0	0
7	S	LPD	0.00	0	0	0	0	0	0	0	0	0	0
8	A	ENG	0.40	0	0	0	2	0	0	2	5	0	0
9	G	INF	0.10	1	0	0	10	2	2	10	1	0	0
10	A	SD	0.40	5	0	0	0	0	0	0	0	0	0
11	A	AH1	0.40	3	4	0	0	6	10	0	1	0	0
12	A	CAS	0.40	1	3	0	0	10	8	0	1	0	0
13	A	VF	0.45	6	1	0	0	1	1	0	0	0	0
14	A	MED	0.30	0	0	0	0	0	0	0	0	10	0
15	S	SMC	0.20	0	0	0	0	0	0	0	10	0	0
16	A	TARP	0.50	0	0	0	0	0	0	0	0	0	6
17	S	AAAV	0.25	0	0	0	0	0	0	0	0	0	0
18	A	MV22	0.45	0	0	0	0	0	0	0	0	0	0
19	A	SAT	0.70	0	0	0	0	0	0	0	0	0	6
20	G	SOF	0.25	0	0	0	6	6	0	6	1	0	10
21	A	BASE	0.00	0	0	0	0	0	0	0	0	0	0

ID	TASK	#	TYPE	LOCATION	COMMENTS
0	HILLS	1	M	(25,60)	prerequisite for north beach
1	AIRPRT	1	M	(5,95)	
2	SEAPRT	1	M	(25,45)	
3	HOLD	3	M	above 3	spawned by tasks 0,1,2
4	BEACH	2	M	(28,73); (28,83)	
5	ARTY	20	D	mostly(10,40)-(20,90)	target PZs 0-4 some out of range of

DDG and FFG

6 FROG	10	D	land area	target N & S beaches
7 SILK(H)	5	D	(28,51or66or89)	2 target CV; 3 target ARG
8 GMINE	4	E	2 each on roads to port & airfield (one is nearby each)	
9 SMINE	8	E	2 before each beach	prereqs for beach tasks 4 "drifting at sea" within 20-30 miles of shore
10 AIR(Sea)	12			6 target each of CV and ARG; vel = 0.4
11 AIR(Gnd)	3+	D		target PZs 0,2-4; vel = 0.2
12 SAM(H)	3	E	2 sites at port, 1 at airport; prereqs to tasks 1 & 2	
13 SAM(N)	*3*	*E*	*2 at port, 1 at airport*	
14 TANK(H)	5	E	2 on North & South roads, 1 at port; vel = 0.02	
15 SILK(N)	*10*	*D*	*(28,51or66or89)*	*6 in North area; 4 in South*
16 SEA(Pb)	8	D	2 target each of PZs 5-8; vel = 0.10	
17 SEA(Sub)	6	D	3 target each of CV and ARG; vel = 0.05	
18 SEA(N)	20?		stuff around to add clutter/confusion	
19 SEA(PbN)	*16*	*D*	*intermixed with tasks of class 16; vel = 0.10*	
20 MEDVC	6	M	spawned by/at tasks of class 1,2,4; 2 spawned elsewhere	
21 AIR(N)	16	D	stuff to add clutter/confusion with class 10	
22 MISSLE ??		unstoppable, spawned by silkworm and/or frog disappears		
23 S. Beach				
24 GTL(H)	1	E?	~(5,60) spawns task 27 on attack; vel = 0.02	
25 GTN(N)	*6*	*E?*	*~(5,60) 3 on each bridge approach; vel =0.02*	
26 BRIDGE	1	M	~(5,60) donít do this bridge!	
27 BRIDGE	1	M	~(5,60) demolition task with time window 120(?)sec	

NOTE-1: If threats of classes 5, 6, 7, are not killed in their opportunity windows (~300sec, 320sec, 600sec, respectively) an unstoppable missile or artillery will impact some PZ.

NOTE-2: Ships should avoid "stumbling" into tasks of classes 16 and 17

NOTE-3: Not attacking task 24 before it reaches a bridge location will spawn a missile launched at ARG. Likewise, for not attacking task 27 in time.

NOTE-4: The appearance of task 24 may not occur until ~ 1000-1500 sec into game. Up to then it will be necessary to sort out tasks of 25 from 24. Also, SAMs (12-13) will not appear until the latter 1/2 of game (as INF approaches PORT and A/P).

APPENDIX F. SAMPLE DEPENDENT VARIABLE FILE, DATA CODING SCHEME, AND DATA TABLE

Appendix F is a sample of the output from the dependent variable file created by DDD and used by the Lead Team in their analyses. Descriptive annotations are show in italics.

A. SAMPLE RAW DATA FILE

Team name: X
Experiment condition: A1post1
Number of tasks arrived: 139
Number of DMs: 4
Number of task classes: 28
Number of penetration zones: 9

Number of task arrivals by task class
{This column depicts the number of times a given task occured. In each of the following data segments there will be a list of 28 values. In each case, that column represents the tasks as labelled below, in the same order.}

1 ← *Hill*
1 ← *Airport*
1 ← *Seaport*
2 ←*Hold*
1 ←N. *Beach*
18 ←*Arty*
8 ←*Frogs*
5 ←*Silkworm*
4 ←*Land Mines*
9 ←*Sea Mines*
11 ←*Air*
3 ←*Helos*
3 ←*SAMs*
3 ←*Dummy SAMs*
4 ←*Ground*
5 ←*Dummy Silkworm*
6 ←*PB*
3 ←*Subs*
0 ←*Neutral Sea*
14 ←*Neutral Sea*
4 ←*Medivac*
18 ←*Neutral Air*
4 ←*Missile*
1 ←*S.Beach*
1 ←*Lead Veh*
7 ←*Neutral Bridge Traffic*
1 ←*OP*
1 ←*Blow Bridge*

Number of initiated attacks by each dm on various task classes
{Each column represents a DM. In this case, there were only 4 DMs (DM0-DM3)
{For each grouping of numbers that follow, the task are always represented in the same order, as depicted below.}

0 0 0 1 ← Hill
0 0 0 1 ← Airport
0 0 0 0 ← Seaport
0 0 0 1 ←Hold
0 0 0 1 ←N. Beach
1 13 2 1 ←Arty
0 6 0 1 ←Frogs
2 0 0 0 ←Silkworm
0 0 2 0 ←Land Mines
6 0 0 0 ←Sea Mines
0 10 0 0 ←Air
0 3 0 0 ←Helos
2 0 0 0 ←SAMs
0 0 0 0 ←Dummy SAMs

3 0 0 1 ←Ground	0 0 0 0 ←Neutral Air
0 0 0 0 ←Dummy Silkworm	0 0 0 0 ←Missile
0 0 0 0 ←PB	0 0 1 0 ←S. Beach
0 2 0 0 ←Subs	0 0 0 0 ←Lead Veh
0 0 0 0 ←Neutral Sea	0 0 0 0 ←Neutral Bridge Traffic
0 0 0 0 ←Neutral Sea	0 0 0 0 ←OP
0 0 0 0 ←Medivac	1 0 0 0 ←Blow Bridge

Number of assisted attacks by each dm on various task classes
{This column represents assist by other DMs in a given attack. If an assist is shown for the initial attacker, this was not counted as an assist in Lead Team calculations}

1 0 0 0	0 0 0 0
0 0 1 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0

Avg accuracy of attacks by each dm on various task classes
{999.00 means that a particular DM did not participate in a task, otherwise the number represents their accuracy.}

999.00 999.00 999.00 73.50	64.00 999.00 999.00 64.00
999.00 999.00 999.00 74.19	999.00 999.00 999.00 999.00
999.00 999.00 999.00 999.00	999.00 999.00 999.00 999.00
999.00 999.00 999.00 100.00	999.00 50.50 999.00 999.00
999.00 999.00 999.00 34.94	999.00 999.00 999.00 999.00
100.00 100.00 100.00 100.00	999.00 999.00 999.00 999.00
999.00 100.00 999.00 100.00	999.00 999.00 999.00 999.00
50.00 999.00 999.00 999.00	999.00 999.00 999.00 999.00
999.00 999.00 100.00 999.00	999.00 999.00 999.00 999.00
100.00 999.00 999.00 999.00	999.00 999.00 100.00 999.00
999.00 80.80 999.00 999.00	999.00 999.00 999.00 999.00
999.00 100.00 999.00 999.00	999.00 999.00 999.00 999.00
50.00 999.00 999.00 999.00	999.00 999.00 999.00 999.00
999.00 999.00 999.00 999.00	51.56 999.00 999.00 999.00

Number of contacts (collisions) by each dm on various task classes
{Not user for Lead Team calculations}

0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
1 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0

Total Number of contacts (collisions): 2

Number of penetrations on PZ's by task classes
{Not user for Lead Team calculations}

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 2 2
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 3 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

Total Number of penetrations: 12

Number of attacks on various task classes

1	17
1	7
0	2
1	2
1	6

10	0
3	0
2	0
0	0
4	1
0	0
0	0
2	0
0	1

Total Number of attacks: 61

Average attack latency time on various task classes
{Not user for Lead Team calculations}

635.00	522.25
2576.00	999.00
999.00	999.00
68.00	787.50
774.00	999.00
60.94	999.00
30.00	999.00
377.75	999.00
1548.25	999.00
665.17	1248.50
62.00	999.00
74.00	999.00
668.25	999.00
999.00	1181.00

B. DATA CODING SCHEME

The following page shows a tabulated summary of the experiment results for each team and trial. The following coding scheme was used to distinguish various data in the MINITAB data table (section C).

Team:

A	→	1
B	→	2
C	→	3
D	→	4
E	→	5
F	→	6
X	→	7
Y	→	8
Z	→	9

Architecture:

A0 Pre Trigger	→	0
A0' Pre Trigger	→	1
A0 Post Trigger	→	2
A0' Post Trigger	→	3
A1	→	4
A2	→	5

The choice column refers to whether a team played their choice architecture, or whether they played an alternative architecture during a particular session.

Choice
1 → Choice
2 → Alternative
3 → Pre Trigger (A0)

There are two columns for each of the 5 mission tasks analyzed in the scenario. The first column (e.g. Hill, etc.) represents the number of players used to complete the task, and the second number (e.g. Hill Acc, etc.) is DDD's accuracy calculation for that task.

Some teams played their choice scenario as their first post trigger run, and some teams played it as their second post trigger run. The table below shows how to intepret the numbers in the choice column on the results table.

Order
1 → Pre Trigger Run
2 → 1st Post Trigger Run
3 → 2nd Post Trigger Run

C. DATA TABLE

Team	Arch	Date	Choice	Order	Hill	Hill Acc	Airport	Apt Acc	Seaport	SptAcc	Beach	Beach Acc	Bridge	Brdg Acc	Hill Val	Apt Val	Spt Val	Bch Val	Brdg Val
6	0	12	3	3	3	100	0	0	0	0	3	100	1	25	10	30	30	10	15
8	0	13	3	3	1	56	0	0	0	0	1	19	0	0	*	*	*	*	*
1	0	13	3	3	2	69	1	66	0	0	0	0	0	0	*	*	*	*	*
2	1	13	3	3	2	100	2	72	0	0	2	77	0	0	*	*	*	*	*
3	0	13	3	3	2	92	1	66	2	96	1	57	0	0	*	*	*	*	*
7	1	14	3	3	2	96	0	0	0	0	2	100	1	51	*	*	*	*	*
4	1	14	3	3	3	100	3	100	0	0	3	100	1	100	*	*	*	*	*
9	0	14	3	3	2	100	2	100	0	0	1	34	1	68	*	*	*	*	*
5	0	14	3	3	2	100	2	100	0	0	3	100	1	51	*	*	*	*	*
7	3	17	1	1	2	77	0	0	0	0	2	76	1	51	*	*	*	*	*
6	4	17	2	1	2	58	0	0	0	0	2	46	0	0	*	*	*	*	*
2	3	17	1	1	3	100	3	91	2	100	1	19	2	98	*	*	*	*	*
1	5	18	2	1	2	80	2	98	0	0	0	0	1	25	*	*	*	*	*
4	5	18	2	1	3	100	3	100	0	0	3	100	1	40	*	*	*	*	*
8	4	19	2	1	1	19	0	0	1	66	1	19	0	0	*	*	*	*	*
5	2	19	1	1	3	100	1	60	3	100	3	100	0	0	*	*	*	*	*
3	2	19	1	1	3	96	3	98	3	100	2	82	0	0	*	*	*	*	*
9	2	20	1	1	2	100	2	100	2	75	2	100	1	40	*	*	*	*	*
2	4	20	2	2	1	80	1	75	1	100	2	94	1	25	*	*	*	*	*
3	5	21	2	2	3	100	3	98	4	100	3	93	1	40	*	*	*	*	*
9	5	21	2	2	1	31	2	98	2	94	2	73	0	0	*	*	*	*	*
7	4	21	2	2	2	73	2	74	0	0	1	34	1	51	*	*	*	*	*
5	4	24	2	2	2	100	0	0	3	76	3	96	1	51	*	*	*	*	*
8	2	24	1	2	1	19	2	63	3	100	1	34	0	0	*	*	*	*	*
6	2	24	1	2	2	62	2	98	2	92	3	93	2	65	*	*	*	*	*
4	3	24	1	2	3	100	2	100	2	75	3	100	1	64	*	*	*	*	*
1	1	25	1	2	2	80	2	75	3	99	2	62	0	0	*	*	*	*	*

LIST OF REFERENCES

Cohen, W. S., *SecDef's Annual Report to the President and the Congress*, p. i, 1998)

Devore J. L., *Probability and Statistics for Engineering and the Sciences*, Fourth ed., Duxbury Press, 1995.

Drake, J. F., "Design and Development of the Scenario for the Second NPS A2C2 Experiment", Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1997.

Joint Warfighting Center, *Concept for Future Joint Operations: Expanding Joint Vision 2010*, May, 1997.

Kleinman, D.L., P.W. Young and G. Higgins, "The DDD-III: A Tool for Empirical Research in Adaptive Organizations", *Proc. 1996 Symposium on Command and Control Research and Technology*, NPS, Monterey, CA, June 1996.

Levchuck, Y.N., Pattipati, K.R., and Curry, M.L., "Normative Design of Organizations to Solve a Complex Mission: Theory and Algorithms", *Proc. 1997 Symposium on Command & Control Research* , Washington, DC, June 17-20, 1997. [Best Paper Award]

Pattipatti, K. R., and others, "*Designing Architectures: Model Predictions versus Experimental Data Analysis*", Brief to A2C2 Research Group, February 1998.

Serfaty,D., "*The A2C2 Project: Overview of Progress*", Brief to A2C2 Research Group, February 1998.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Road
   Quantico, VA 22134-5027

4. Director, Marine Corps Research Center 2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, VA 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC, Code C45
   3300 Russell Road
   Quantico, VA 22134-5130

6. Marine Corps Representative 1
   Naval Postgraduate School
   Code 037
   Monterey, CA 93943

7. Marine Corps Tactical Systems Support Activity 1
   Technical Advisory Branch
   Attn: Maj J.C. Cummiskey
   Box 555171
   Camp Pendleton, CA 92055-5080

8. Chairman, C4I Academic Group 1
   Code CC/Bo
   Naval Postgraduate School
   Monterey, CA 93943

9. Professor William G. Kemple 1
   Code CC/Ke
   Naval Postgraduate School
   Monterey, CA 93943

10. Professor Gary R. Porter.. 1
Code CC/Po
Naval Postgraduate School
Monterey, CA 93943

11. Capt Robert E. L. Benson, USMC... 2
c/c Maj J.C. Cummiskey
Box 555171
Camp Pendleton, CA 92055-5080